UNIVERSITY OF CALIFORNIA

Los Angeles

Gaining Justified Human Trust by Improving Explainability in Vision and Language Reasoning Models

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Arjun Reddy Akula

2021

ABSTRACT OF THE DISSERTATION

Gaining Justified Human Trust by Improving Explainability in Vision and Language Reasoning Models

by

Arjun Reddy Akula
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2021
Professor Song-Chun Zhu, Chair

In recent decades, artificial intelligence (AI) systems are becoming increasingly ubiquitous from low risk environments to high risk environments such as chatbots, medical-diagnosis and treatment, self-driving cars, drones and military applications. However understanding the behavior of AI systems built using black box machine learning (ML) models such as deep neural networks remains a significant challenge as they cannot explain why they reached a specific recommendation or a decision. Explainable AI (XAI) models, through explanations, address this issue by making the underlying inference mechanism of AI systems transparent and interpretable to expert users (system developers) and non-expert users (end-users). Moreover, as the decision making is being shifted from humans to machines, transparency and interpretability achieved with reliable explanations is central to solving AI problems such as *safely operating self-driving cars*, *detecting and mitigating bias in machine learning (ML) models*, *increasing justified human trust in AI models*, *efficiently debugging models*, and *ensuring that ML models reflect our values*. In this thesis, we propose new methods to effectively gain human trust in vision and language reasoning models by generating adaptive and human understandable explanations and also by improving interpretability, faithfulness,

and robustness of the existing models. Specifically, we make the following four major contributions:

- First, motivated by Song-Chun Zhu's work on generating abstract art from photographs, we pose **explanation as a procedure/path** to explain the image interpretation, i.e. a parse graph. Also, in contrast to the current methods in XAI that generate explanations as a single shot response, we pose explanation as an iterative communication process, i.e. dialog, between the machine and human user. To do this, we use **Theory of Mind (ToM)** which helps us in explicitly modeling human's intention, machine's mind as inferred by the human as well as human's mind as inferred by the machine. In other words, these explicit mental representations in ToM are incorporated to learn an optimal explanation path that takes into account human's perception and beliefs. We call this framework **X-ToM**. We show that the mental representations in ToM help in quantitatively measuring and increasing justified human trust in the machine. We present applications of the proposed approach to three visual recognition tasks, namely, image classification, action recognition, and human body pose estimation. We argue that our ToM based explanations are practical and more natural for both expert and non-expert users to understand the internal workings of complex machine learning models. Extensive human study experiments verify our hypotheses, showing that the proposed explanations significantly outperform the state-of-the-art XAI methods in terms of all the quantitative and qualitative XAI evaluation metrics including human trust, reliance, and explanation satisfaction.

- We propose a Conceptual and Counterfactual Explanation framework, which we call **CoCo-X**, for explaining decisions made by a deep convolutional neural network (CNN). In Cognitive Psychology, the factors (or semantic-level features) that humans zoom in on when they imagine an alternative to a model prediction are often referred to as ***fault-lines***. Motivated by this, our CoCoX model explains decisions made by a CNN using fault-lines. Specifically, given an input image I for which a CNN classification model M predicts class c_{pred}, our fault-line based explanation identifies the minimal semantic-level features (e.g., *stripes* on zebra, *pointed ears* of dog), referred to as explainable concepts, that need to be added to or deleted from I in order to alter the classification category of I by M to another specified class c_{alt}.

- In addition to proposing explanation frameworks such as X-ToM and CoCo-X, we also evaluate existing deep learning models such as Transformer, Compositional Modular Networks in terms of their ability to provide interpretable visual and language representations and their ability to provide robust predictions to out-of-distribution samples. We show that the state-of-the-art end-to-end modular network implementations - although provide high model interpretability with their transparent, hierarchical and semantically motivated architecture - require a large amount of training data and are less effective in generalizing to unseen but known language constructs. We propose several extensions to modular networks that mitigate bias in the training and improve robustness and faithfulness of model;
- The research culminates in a visual question and answer generation framework, in which we propose a semi-automatic framework for generating out-of-distribution data to explicitly understand the model biases and help improve the robustness and fairness of the model.

The dissertation of Arjun Reddy Akula is approved.

Yingnian Wu

Hongjing Lu

Kai-Wei Chang

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2021

To my parents, brother and sister . . .

for their unconditional love and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost I'd like to express my deepest gratitude towards my advisor Prof. Song-Chun Zhu for his continuous, invaluable support and encouragement throughout my Ph.D. program. I am deeply impressed and influenced by his visionary ideas, expertise, and guidance and this work wouldn't have been possible without him. Especially his diverse and in-depth knowledge across all the areas of AI such as computer vision, robotics, language processing, cognitive science inspired me and helped me grow as a researcher. When I was applying for my PhD program in 2015, he was the only professor among all the top USA universities and research labs who was working on such diversified topics and long-term challenges in AI. That is also the primary reason for me to explore multi-modal reasoning approaches as part of my Ph.D. research. I also thank him for providing super strong recommendation letters and support, and for the opportunity to travel and have friendly, insightful conversations with him during multiple DARPA XAI PI visits, annual lab outings, parties at his home and other conferences. I am really fortunate to have the best advisor I can ever imagine.

I am also deeply grateful to Prof. Joyce Chai (UMich), for her excellent co-guidance in NLP methodology and experiment design, and also for improving my ability in conducting reproducible research and in writing, publishing high-quality papers and journals. I thank Prof. Sinisa Todorovic (OSU) for his invaluable support and guidance in shaping my research work, organizing reports and submitting demonstrations to the DARPA XAI program.

I thank Prof. Yingnian Wu, Prof. Hongjing Lu and Prof. Kai-Wei Chang for serving on my thesis committee and for their tremendous support, guidance and valuable suggestions, discussions in conducting several key experiments and human studies as part of my research work.

I also thank Prof. Siva Reddy (Mila), Dr. Spandana Gella (Amazon Alexa AI), Dr. Changsong Liu, Dr. Radu Soricut (Google Research), Prof. Mohit Bansal (UNC), Prof. Jesse Thomason (USC), Dr. Varun Jampani (Google Research) for excellent collaborations and discussions along the way. I also extend my thanks to Dr. Soravit Changpinyo, Piyush Sharma, Dr. Boqing Gong, Dr. Aishwarya Padmakumar, Mahdi Namazifar and Dr. Xiaodan Liang for sharing their knowledge and guiding

me on my work.

My appreciation and thanks to all the members in UCLA for their friendship, collaboration, and support throughout my time here: Dr. Keze Wang, Dr. Yixin Zhu, Dr. Yuanlu Xu, Dr. Tao Yuan, Dr. Mark Edmonds, Dr. Mitch Hill, Dr. Hang Qi, Dr. Hangxin Liu, Dr. Siyuan Qi, Dr. Nishant Shukla, Dr. Lifeng Fan, Feng Gao, Dr. Ruiqi Gao, Xiaofeng Gao, Qing Li, Jonathan Mitchell, Erik Nijkamp, Feng Shi, Luyao Yuan, Yizhou Zhao, Zilong Zheng, Dr. Yujia Peng, Sari Saba-Sadiya, Yixin Chen, Steven Gong, Shu Wang, Lawrence Chen, Dr. Xu Xie, Dr. Zack Stokes, Pradeep Dogga, Siva Kesava Reddy Kakarla, Ashutosh Kumar, Murali Ramanujam, and Aishwarya Sivaraman. Special thanks to my best friend Mahesh Goud Tandarpally for unconditional support and chats about research, career and life.

I would like to thank Prof. Hongquan Xu, Glenda Jones, Laurie Leyden, Chie Ryu for their outstanding support in administrative and general matters. I sincerely acknowledge the support and facilities offered by the Department of Statistics at UCLA.

Finally, I dedicate this dissertation to my parents, my brother Dr. Aneesh Reddy Akula and my sister Dr. Praveena Reddy Akula for their support and unconditional love. Special thanks to my brother for providing a constant source of support and advice through out my bachelors, masters, and doctoral studies.

This work was supported by DARPA XAI N66001-17-2-4029, ARO W911NF1810296 and ONR MURI N00014-16-1-2007. I am grateful for the collaboration, inspiration, and funding support these grants provided.

VITA

2016-2021 Graduate Research Assistant, VCLA@UCLA (DARPA XAI Program), Prof. Song-Chun Zhu.

2021 [Summer 2021] Applied Scientist Intern, Amazon Alexa AI, Dr. Spandana Gella, Prof. Mohit Bansal, and Dr. Dilek Hakkani-Tur.

2021 [Spring & Fall 2021] Teaching Assistant, Statistics Department, UCLA.

2020 [Summer 2020] Research Intern, Google Research, Dr. Radu Soricut.

2019 [Summer 2019] Applied Scientist Intern, Amazon AI, Dr. Spandana Gella, Prof. Siva Reddy, and Dr. Yaser Al-Onaizan.

2018 Ph.D. Candidate in Statistics, UCLA.

2014-2016 Research Software Engineer, IBM Research AI, India.

2012-2014 MS by Research in Computer Science and Engineering, IIIT Hyderabad, India.

2008-2012 B.Tech in Computer Science and Engineering, IIIT Hyderabad, India.

PUBLICATIONS

CX-ToM: Counterfactual Explanations with Theory-of-Mind for Enhancing Human Trust in Image Recognition Models. **Arjun Akula**, Keze Wang, Changsong Liu, Sari Saba-Sadiya, Hongjing Lu, Sinisa Todorovic, Joyce Chai, Song-Chun Zhu. *iScience Cell Press Journal* 2021.

Robust Visual Reasoning via Language-Guided Neural Module Networks. **Arjun Akula**, Varun Jampani, Soravit Changpinyo, Song-Chun Zhu. *NeurIPS*, 2021.

Contextual Neural Module Networks for Grounding Visual Referring Expressions. **Arjun Akula**, Spandana Gella, Keze Wang, Song-Chun Zhu, Siva Reddy. *EMNLP*, 2021 (Long Paper, Main).

CrossVQA: Generating Scalable and Nonstationary Benchmarks for Testing VQA Generalization. **Arjun Akula**, Soravit Changpinyo, Boqing Gong, Piyush Sharma, Song-Chun Zhu, Radu Soricut. *EMNLP*, 2021 (Oral, Long Paper, Main).

CoCoX: Generating Conceptual and Counterfactual Explanations via Fault-Lines. **Arjun Akula**, Wang Shuai, Song-Chun Zhu. *AAAI*, 2020 (Oral, Spotlight, acceptance rate: 20.5%).

Words aren't enough, there order matters: On the Robustness of Grounding Visual Referring Expressions. **Arjun Akula**, Song-Chun Zhu. *Association for Computational Linguistics (ACL)*, 2020 (Oral, acceptance rate: 17.6%).

Visual Discourse Parsing. **Arjun Akula**, Wang Shuai, Song-Chun Zhu. *CVPR 2019 workshop on Language and Vision* (Oral).

Natural Language Interaction with Explainable AI models. **Arjun Akula**, Sinisa Todorovic, Joyce Chai, Song-Chun Zhu. *CVPR 2019 workshop on Explainable AI.*

Explainable AI as Collaborative Task Solving. **Arjun Akula**, Changsong Liu, Sinisa Todorovic, Joyce Chai, Song-Chun Zhu. *CVPR 2019 workshop on Explainable AI.*

CHAPTER 1

Introduction

1.1 Motivation and Objective

The motivation of this dissertation is to propose effective methods to gain justified human trust in AI models by providing users with adaptive and human-friendly explanations. As part of this high-level goal, we also critically examine the existing models to understand their biases and also propose new methods to mitigate biases and improve robustness, interpretability and fairness of the models.

1.1.1 Importance of Explanations

Artificial Intelligence (AI) systems are becoming increasingly ubiquitous from low risk environments such as movie recommendation systems and chatbots to high risk environments such as medical-diagnosis and treatment, self-driving cars, drones, IT support and military applications [CBP15, GPC16, LCW17, MKS13, PAS13, DNA14, AAA17, AAD18, GAD16, ADE21, ADN18, BRH17, ADE21]. In particular, AI systems built using black box machine learning (ML) models – such as deep neural networks and large ensembles [Lip16, RSG16, Mil18a, YGS18, STY17, RAR16, ZF14, STK17, KRS14] – perform remarkably well on a broad range of tasks and are gaining widespread adoption. However understanding the behavior of these systems remains a significant challenge as they cannot explain why they reached a specific recommendation or a decision. This is especially problematic in high risk environments such as banking, healthcare, and insurance, where AI decisions can have significant consequences. Therefore, much hope rests on explanation methods as tools to understand the decisions made by these AI systems.

Explainable AI (XAI) models, through explanations, make the underlying inference mechanism of AI systems transparent and interpretable to expert users (system developers) and non-expert users (end-users) [Lip16, RSG16, Mil18a, Hof17b, Lip90, SK11]. Explanations play a key role in integrating AI machines into our daily lives, i.e. XAI is essential to increase social acceptance of AI machines (see Figure 1.1). As the decision making is being shifted from humans to machines, **transparency** and **interpretability** achieved with reliable explanations is central to solving AI problems such as the following:

1. Safety [MBT19] (e.g. *How to operate self-driving cars safely?*)
2. Bias & Fairness [BDH18] (e.g. *How to detect and mitigate bias in ML models?*)
3. Justified Human Trust in ML models [SW18] (e.g. *How to trust the output of AI systems to inform our decisions?*)
4. Model Debugging [Hal19] (e.g. *How to improve my model by identifying points of model failure?*)
5. Ethics [VKK19] (e.g. *How to ensure that ML models reflect our values?*)

1.2 Measuring Justified Human Trust

In this dissertation, we focus on two dimensions of trust: **Justified Positive Trust** (JPT) and **Justified Negative Trust** (JNT) [HMK18]. We measure JPT and JNT by evaluating the human's understanding of the machine's (M) decision-making process. For example, let us consider an image classification task. Suppose if the machine M predicts images in the set C correctly and makes incorrect decisions on the images in the set W. Intuitively, JPT will be computed as the percentage of images in C that the human subject felt M would correctly predict. Similarly, JNT (also called as mistrust), will be computed as the percentage of images in W that the human subject felt M

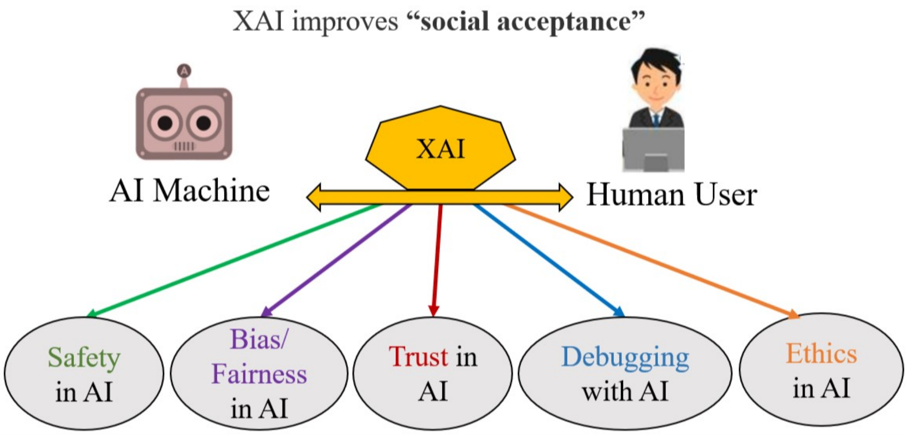


Figure 1.1: An AI machine that explains its predictions to human users will find more social acceptance. Therefore, XAI models are the key in addressing the issues such as Safety in AI, Bias/Fairness in AI, Trust in AI, Model Debugging, and Ethics in AI.

would fail to predict correctly. In other words, given an image, justified trust evaluates whether the users could reliably predict the model’s output decision. Note that this definition of justified trust is domain generic and can be applied to any task. For example, in an AI-driven clinical world, our definitions of JPT and JNT can effectively measure how much doctors and patients understand the AI systems that assist in clinical decisions.

1.3 Limitations in the existing XAI models

We identify the following two key limitations in the state-of-the explainable AI models:

1. **Attention is not a Good Explanation:** Previous studies have shown that trust is closely and positively correlated to the level of how much human users understand the AI system — *understandability* — and how accurately they can predict the system’s performance on a given task — *predictability* [Hof17b, Lip16, HMK18, Mil18a]. Therefore there has been a growing interest in developing explainable AI systems (XAI) aimed at increasing

understandability and predictability by providing explanations about the system's predictions to human users [Lip16, RSG16, Mil18a, YGS18]. Current works on XAI generate explanations about their performance in terms of, e.g., feature visualization and attention maps [STY17, RAR16, ZF14, STK17, KRS14, ZNZ18]. However, solely generating explanations, regardless of their type (visualization or attention maps) and utility, *is not sufficient* for increasing understandability and predictability [JW19]. We verify this in our experiments.

2. **Explanation is an Interactive Communication Process:** Existing methods for XAI generate explanations as a single shot response. We believe that an effective explanation cannot be one shot and involves iterative process of communication between the human and the machine. The context of such interaction plays an important role in determining the utility of the follow-up explanations [CS89]. As humans can easily be overwhelmed with too many or too detailed explanations, interactive communication process helps in understanding the user and identify user-specific content for explanation. Moreover, cognitive studies [Mil18a] have shown an explanation can only be optimal if it is generated by taking user's perception and belief into account.

1.4 Research questions and contributions

In this dissertation, we address the limitations of the existing XAI frameworks and propose new methods to make the AI models more transparent and understandable. We advance the state of the art in XAI, vision and language grounding, and interpretable neural architectures to achieve these goals. Below, we discuss the concrete research questions that we address in this thesis and our key contributions:

Research question 1: *Given a visual recognition task, there exists multiple explanations to justify the model's underlying reasoning process. How to generate user-specific adaptive explanations without overwhelming user with too many or too detailed explanation?*

Song-Chun Zhu, a Chinese mathematician, statistician, and computer scientist, proposed multiple models and algorithms for the rendering of abstract paintings that is capable of controlling the entropy to the user's desired levels [ZZ13]. Extending Zhu's work to the explanation generation framework, we pose explanation as a procedure/path to explain the image interpretation, i.e. a parse graph. Specifically, we learn an optimal explanation path that takes into account human's perception and beliefs. In Chapter 2, we do this by introducing an interactive explanation framework, **X-ToM**. In our framework, the machine generates sequence of explanations in a dialog which takes into account three important aspects at each dialog turn: (a) human's intention (or curiosity); (b) human's understanding of the machine; and (c) machine's understanding of the human user. To do this, we use Theory of Mind (ToM) which helps us in explicitly modeling human's intention, machine's mind as inferred by the human as well as human's mind as inferred by the machine. The ability to reason about other's perception and beliefs, in addition to one's own perception and beliefs, is often referred to as the Theory-of-Mind [DA16, Gol12, PW78].

More specifically, in X-ToM, the machine and the user are positioned to solve a collaborative task, but the machine's mind (M) and the human user's mind (U) only have a partial knowledge of the environment (see Figure 1.2). Hence, the machine and user need to communicate with each other, using their partial knowledge, otherwise they would not be able to optimally solve the collaborative task. The communication consists of two different types of question-answer (QA) exchanges — namely, a) Factoid question-answers about the environment (W-QA), where the user asks "WH"-questions that begin with `what`, `which`, `where`, and `how`; and b) Explanation seeking question-answers (E-QA), where the user asks questions that begin with `why` about the machine's inference. At each turn in the collaborative dialog, our X-ToM updates a model of human perception and beliefs, and uses this model for generating a maximum utility explanation that (a) minimizes the total number of explanations in the dialog and the explanation content; and (b) maximizes user's understandability and predictability about the machine's predictions.

We show that the mental representations in ToM help in quantitatively measuring and increasing justified human trust in the machine. We present applications of the proposed approach to three visual recognition tasks, namely, image classification, action recognition, and human body pose

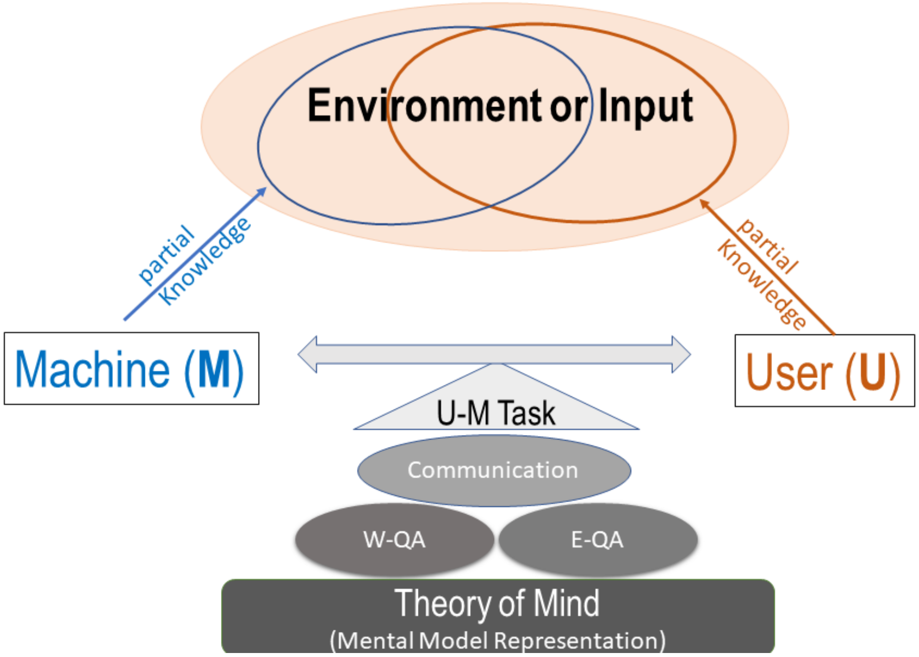


Figure 1.2: **XAI as Collaborative Task Solving**: Our interactive and collaborative XAI framework based on the Theory of Mind. The interaction is conducted through a dialog where the user poses questions about facts in the environment (W-QA) and explanation seeking questions (E-QA).

estimation. We argue that our ToM based explanations are practical and more natural for both expert and non-expert users to understand the internal workings of complex machine learning models. Extensive human study experiments verify our hypotheses, showing that the proposed explanations significantly outperform the state-of-the-art XAI methods in terms of all the quantitative and qualitative XAI evaluation metrics including human trust, reliance, and explanation satisfaction.

Research question 2: *Humans do not explain their understanding through pixels or saliency maps. Instead, they explain through high-level semantic concepts. Is it possible to produce such human-level explanations?*

In Chapter 3, we propose a novel conceptual and counterfactual explanation framework, which we call **CoCo-X**, for explaining decisions made by a deep convolutional neural network (CNN). In Cognitive Psychology, the factors (or semantic-level features) that humans zoom in on when

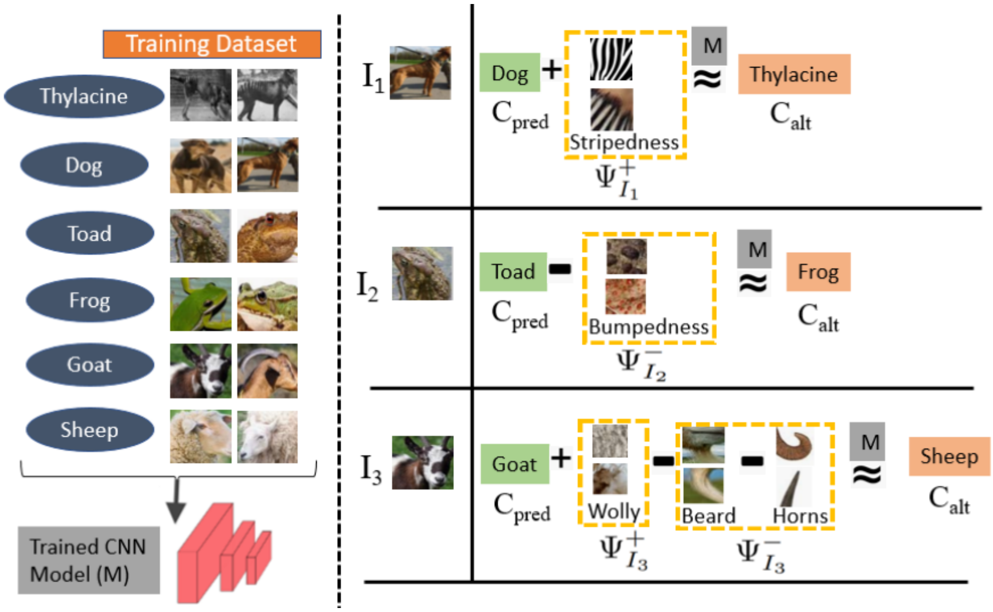


Figure 1.3: **CoCoX Explanations using Fault-Lines**: Positive fault-line explanation ($\Psi^+_{I_1}$) suggests adding *stripes* to the animal in the input image (I_1) to alter the model M's prediction from `Dog` class to `Thylacine` class, i.e., the concept of *stripedness* is critical for M to decide between `Dog` and `Thylacine` in I_1. Similarly, negative fault-line $\Psi^-_{I_2}$ suggests removing *bumps* from I_2 to alter the classification category from `Toad` to `Frog`. Changing the classification result of I_3 from `Goat` to `Sheep` requires adding *wool* and removing *beard* and *horns* from I_3, i.e., it needs both positive and negative fault-lines.

they imagine an alternative to a model prediction are often referred to as ***fault-lines***. Motivated by this, our CoCoX model explains decisions made by a CNN using fault-lines. Specifically, given an input image I for which a CNN classification model M predicts class c_{pred}, our fault-line based explanation identifies the minimal semantic-level features (e.g., *stripes* on zebra, *pointed ears* of dog), referred to as explainable concepts, that need to be added to or deleted from I in order to alter the classification category of I by M to another specified class c_{alt}.

For example, let us consider a training dataset for an image classification task shown in Figure 1.3 containing the classes `Dog`, `Thylacine`, `Frog`, `Toad`, `Goat` and `Sheep`, and a CNN based classification model M which is trained on this dataset. In order to alter the model's prediction of input image I_1 from Dog to Thylacine, the fault-line ($\Psi^{+}_{I_1,c_{pred},c_{alt}}$) suggests adding *stripes* to the Dog. We call this a positive fault-line (PFT) as it involves adding a new xconcept, i.e., *stripedness*, to the input image. Similarly, to change the model prediction of I_2 from Toad to Frog, the fault-line ($\Psi^{-}_{I_2,c_{pred},c_{alt}}$) suggests removing *bumps* from the Toad. We call this a negative fault-line (NFT) as it involves subtracting xconcept, i.e., *bumpedness*, from the input image. In most cases, both PFT and NFT are needed to successfully alter the model prediction.

While there are recent works on generating pixel-level counter-factual and contrastive explanations [HHD18, DCL18, GWE19], to the best of our knowledge, this is the first work to propose a method for generating explanations that are counter-factual as well as conceptual.

We identify two main challenges in generating a fault-line explanation, namely: (a) How to identify the set of xconcepts; and (b) How to select the most critical xconcepts that alter the model prediction from c_{pred} to c_{alt}. In this work, we first propose a novel method to mine all the plausible xconcepts from the given dataset automatically. We then identify class-specific xconcepts by using directional derivatives [KWG18]. Finally, we pose the derivation of a fault-line as an optimization problem which selects a minimal set of these xconcepts to alter the model's prediction. We perform extensive human study experiments to demonstrate the effectiveness of our approach in improving human understanding of the underlying classification model.

Through our human studies, we show that our fault-line based explanations significantly outperform the baselines (i.e., attribution techniques and pixel-level counterfactual explanations) in terms of qualitative and quantitative metrics such as Justified Trust and Explanation Satisfaction [HMK18].

Research question 3: *Recently, several deep learning models have achieved tremendous progress on vision and language grounding datasets. Is it possible to understand the extent to which these models are interpretable, and also verify if these models exploit an unintended biases from the datasets to gain good performance on test sets?*

In Chapter 4, we evaluate existing deep learning models such as Transformer, Compositional Modular Networks in terms of their ability to provide interpretable visual and language representations and their ability to provide robust predictions to out-of-distribution samples. To show our analysis, we consider the task of visual referring expression recognition: a challenging task that requires natural language understanding in the context of an image [KOM14, NMD16, MHT16, HXR16]. To measure the true progress of existing models, we split the existing test sets for this task into two sets, one which requires reasoning on linguistic structure and the other which doesn't. Additionally, we create an out-of-distribution dataset by asking crowdworkers to perturb in-domain examples such that the target object changes. Using these datasets, we empirically show that existing methods fail to exploit linguistic structure and are 12% to 23% lower in performance than the established progress for this task.

In Chapter 5, we show that the state-of-the-art end-to-end modular network (NMNs) implementations [SBG20, AGA20a] - although provide high model interpretability with their transparent, hierarchical and semantically motivated architecture - require a large amount of training data and are less effective in generalizing to unseen but known language constructs. For example, NMNs fail to understand new concepts such as "*yellow sphere to the left*" that are constructed using a combinations of known concepts from train data such as "*blue sphere*", "*yellow cube*", and "*metallic cube to the left*". One of the main reasons for this is that the neural modules in existing works either use a shallow, indirect language guidance [PSV18, HAR17, ASM13] or pre-define the textual inputs in the module instantiation [JHM17b, LLB19], ignoring the rich correlations among the visual inputs and the relevant context from the textual inputs. For example, the neural module that filters based on the object size, "`filter_size(smallest)`", needs to localize a tiny sphere or a medium-sized sphere in the image depending on the object relationships in the expression (e.g. "*the smallest thing among the spheres*" vs. "*the metallic sphere smaller than all the large cylinders*") and the different sizes of spheres and cylinders available in its visual input. We believe that explicitly conditioning the neural modules on the joint textual and visual context helps in inferring robust visiolinguistic relationships which further enhances the compositional reasoning skills. In this dissertation, we propose several extensions to modular networks that mitigate bias in the training and improve robustness and faithfulness of model. The research culminates in a visual question and answer generation framework in Chapter6, in which we propose a semi-automatic framework for generating out-of-distribution data to explicitly understand the model biases and help improve the robustness and fairness of existing models.

CHAPTER 2

Collaborative Explanation with Theory-of-Mind

2.1 Introduction

Explainable AI (XAI) models, through explanations, make the underlying inference mechanism of AI systems transparent and interpretable to expert users (system developers) and non-expert users (end-users) [Lip16, RSG16, Mil18a, Hof17b, Lip90, SK11, ALT19]. Explanations play a key role in integrating AI machines into our daily lives, i.e. XAI is essential to increase social acceptance of AI machines.

Most work on XAI typically focuses on black-box models and generating explanations about their performance in terms of, e.g., feature visualization and attribution [STY17, RAR16, ZF14]. However, solely generating explanations, regardless of their type (visualization or attribution) and utility, *is not sufficient* for increasing understandability and predictability. Previous studies have shown that trust is closely and positively correlated to the level of how much human users understand the AI system — *understandability* — and how accurately they can predict the system's performance on a given task — *predictability* [Hof17b, Lip16, HMK18, Mil18a]. Therefore there has been a growing interest in developing explainable AI systems (XAI) aimed at increasing understandability and predictability by providing explanations about the system's predictions to human users [Lip16, RSG16, Mil18a, YGS18]. Current works on XAI generate explanations about their performance in terms of, e.g., feature visualization and attention maps [STY17, RAR16, ZF14, STK17, KRS14, ZNZ18]. However, solely generating explanations, regardless of their type (visualization or attention maps) and utility, *is not sufficient* for increasing understandability and predictability [JW19]. We verify this in our experiments (see Section 4). To address this issue, We argue that an effective explanation cannot be one shot and involves iterative process of communication between the human and the machine. The context of such interaction plays an important role in determining the utility of the follow-up explanations [CS89]. As humans can easily be overwhelmed with too many or too detailed explanations, interactive communication process helps in understanding the user and identify user-specific content for explanation. Moreover, cognitive studies [Mil18a] have shown an explanation can only be optimal if it is generated by taking user's perception and belief into account.

In our experiments, we found that it is difficult to evaluate the effectiveness of explanations without constraining the communication process. Therefore, we constrain the communication by explicitly defining a collaborative task-solving game for the human user where the effectiveness of the explanations is measured based on the total number of tasks successfully solved by the user and the total number of explanations shown to the user in the communication dialog.

Thus, in this Chapter, we introduce an interactive explanation framework, **X-ToM**. In our framework, the machine generates sequence of explanations in a dialog which takes into account three important aspects at each dialog turn: (a) human's intention (or curiosity); (b) human's understanding of the machine; and (c) machine's understanding of the human user. To do this, we use Theory of Mind (ToM) which helps us in explicitly modeling human's intention, machine's mind as inferred by the human as well as human's mind as inferred by the machine. The ability to reason about other's perception and beliefs, in addition to one's own perception and beliefs, is often referred to as the Theory-of-Mind [DA16, Gol12, PW78, ALS19, AWL21].

More specifically, in X-ToM, the machine and the user are positioned to solve a collaborative task, but the machine's mind (M) and the human user's mind (U) only have a partial knowledge of the environment (see Figure 1.2). Hence, the machine and user need to communicate with each other, using their partial knowledge, otherwise they would not be able to optimally solve the collaborative task. The communication consists of two different types of question-answer (QA) exchanges — namely, a) Factoid question-answers about the environment (W-QA), where the user

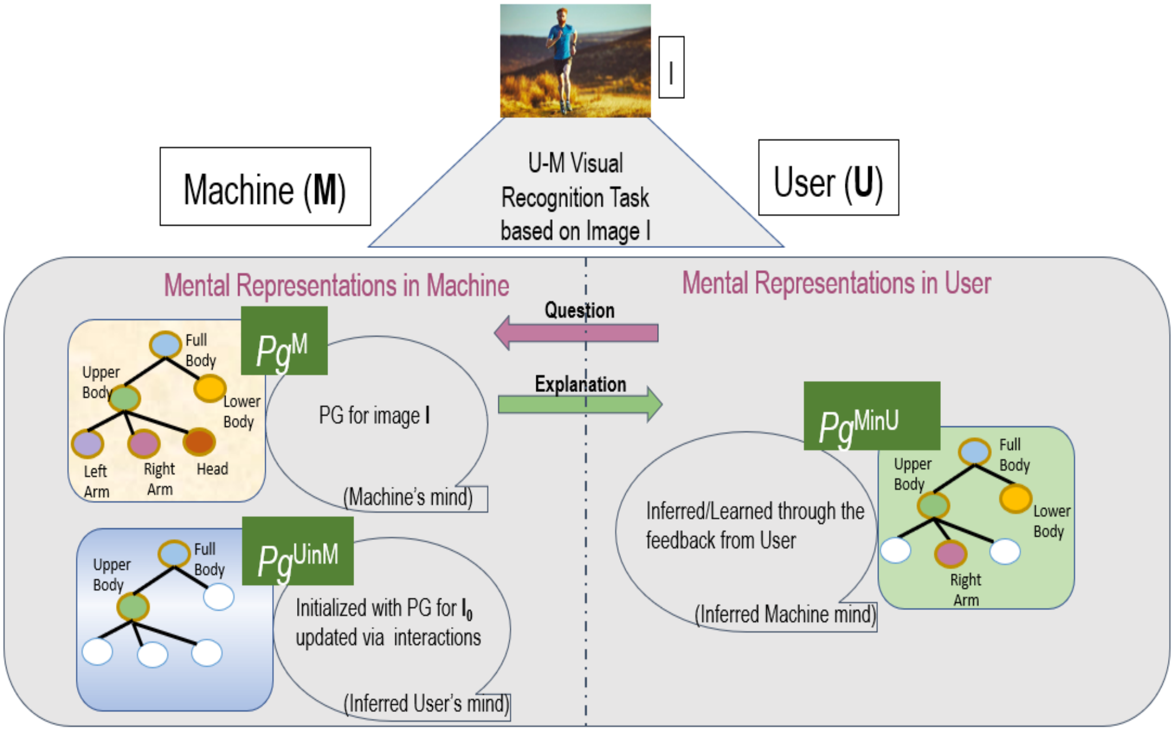


Figure 2.1: X-ToM for a visual recognition task consists of three distinct parse graphs (pg's): pg^M representing the machine's interpretation of the image, pg^{UinM} — the human's mind as inferred by the machine; and pg^{MinU} — the machine's mind as inferred by the human. Nodes of a parse graph represent objects and parts appearing in the image, and edges represent spatial relationships of the objects. X-ToM optimizes explanations so as to reduce a difference among the three parse graphs.

asks "WH"-questions that begin with `what`, `which`, `where`, and `how`; and b) Explanation seeking question-answers (E-QA), where the user asks questions that begin with `why` about the machine's inference. At each turn in the collaborative dialog, our X-ToM updates a model of human perception and beliefs, and uses this model for generating a maximum utility explanation that (a) minimizes the total number of explanations in the dialog and the explanation content; and (b) maximizes user's understandability and predictability about the machine's predictions.

We applied our framework to three visual recognition tasks, namely, image classification, action recognition, and human body pose estimation. In these visual recognition tasks, the machine is given an original image and is supposed to detect and localize objects and parts of interest or a human activity appearing in the image. The user is given a blurred version of the original image, and the user seeks the machine's help essentially through the explanations generated by the machine in order to recognize objects/parts in the blurred image. This provides a unique collaborative setting where the system is motivated to provide human-understandable explanation for its visual recognition and the user is motivated to seek the system's recognition and explanation to help his/her own understanding. To facilitate this collaborative interaction, X-ToM explicitly models mental states of visual understanding ("minds") of the machine and user using parse graphs (pg) in the form of And-Or Graph (AOG) [ZM07]. In a pg, nodes represent objects and parts detected in the image, and edges represent spatial relationships identified between the objects. As shown in Figure 2.1, X-ToM mind models include:

- $\mathbf{pg^{M}}$: the machine's own inference about objects and their locations in the image.
- $\mathbf{pg^{UinM}}$: the human's mind as inferred by the machine.
- $\mathbf{pg^{MinU}}$: the machine's mind as inferred by the human.

These explicit mental representations allow for formalizing the notions of justified trust and mistrust in the machine, as well as quantifying their desired increase through the process of generating explanations.

Using Amazon Mechanical Turk, we have collected explanation dialogs by interacting with turkers through X-ToM framework. From there, X-ToM learned an optimal explanation policy that takes into account user perception and beliefs. Through our extensive human studies, we show that X-ToM allows the user to achieve a high success rate in visual recognition on blurred images, and does so very efficiently in a few dialog exchanges. We also found that the most popularly used attribution based explanations (viz. saliency maps) are not effective to improve human trust in AI system, whereas our Theory-of-Mind inspired approach significantly improves human trust in AI by providing effective explanations.

2.1.1 Contributions

Our contributions in this Chapter are threefold: (i) a new interactive XAI framework based on the Theory-of-Mind; (ii) a new collaborative task-solving game in the domain of visual recognition for learning collaborative explanation strategies; and (iii) a new objective measure of trust and quantitative evaluation of how humans gain increased trust in a given vision system.

2.2 Related Work

The importance of generating explanations or justifications of decisions made by an AI system has been emphasized and widely explored in numerous works over the past decades [Ala17, Bor16, CBS17, BBM15, SGK17, ZKL16, BB87, BC17, Dar13, DK17a, DK17b, GF17, Hof17a, HK17, She17, SM18, TZS16, WKR16, ATC19]. For the rest of this section, we use the term 'Performer' to refer to the AI model that needs to be explained. As shown in Figure 2.2, most prior work in XAI fall into one or more of the following categories:

2.2.1 Intrinsic vs Post-hoc Explanations

Explanations that are derived (or understood) directly from the performer's internal representation or the output parse structure are called as Intrinsic Explanations [DK17b, ZNZ18, ZWZ18, SWS17]. For example, the reasoning behind the predictions made by linear regression models, decision trees, and And-Or Graphs [LHW13, ZCN17] is easier to understand without using any external XAI models and hence are considered as intrinsically explainable. These performers, due to their simple structure, typically do not fare well in terms of performance compared to black-box performers such as deep neural nets. Majority of the work in XAI is focused on generating post-hoc [LBJ16, RSG16, KWG18, KRS14, WRV16, KSD15] explanations where an external XAI model is employed to explain an already trained performer. More recently, there are efforts in making the complex deep neural networks intrinsically explainable [ZWZ18, ZYM19, ZYY18].

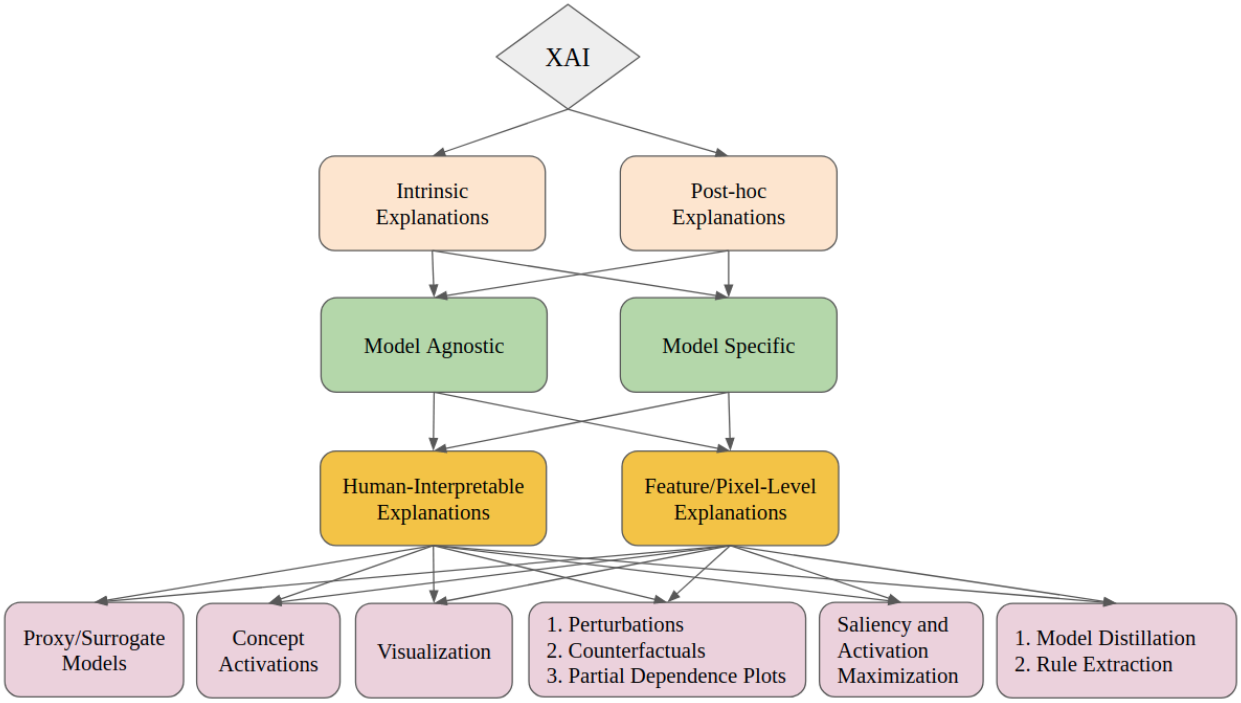


Figure 2.2: Types of AI Explainability

For example, [ZYM19] proposed a decision tree to encode decision modes in fully-connected layers and thereby quantitatively explain the logic for each CNN prediction.

2.2.2 Model-agnostic vs Model-specific Explanations

Explainable AI models that do not require performer specific details (for example, weights of deep neural nets) for generating explanations are called as model-agnostic models [RSG18]. In other words, they simply analyze the dependencies of input features against the output predictions to explain the performer's decision. It may be noted that intrisinc explanations are typically model-specific whereas post-hoc XAI models are model-agnostic. Several XAI works belong to this category, to name a few:

1. *Local Intepretable Model-Agnostic Explanation (LIME)* [RSG16]. LIME produces attention map as explanation, generated through super-pixel based perturbation. Though LIME is a

post-hoc model-agnostic model, it generates explanations by approximating the performer (locally) with an intrinsic model-specific XAI model.

2. *Contrastive Explanation Methods (CEM)* [DCL18]. CEM provides contrastive explanations by identifying pertinent positives and pertinent negatives in the input image.

3. *Counterfactual Visual Explanations (CVE)* [GWE19]. CVE provides counterfactual explanation describing what changes to the situation would have resulted in arriving at the alternative decision.

2.2.3 Human Interpretable Explanations (Concept Activation Vectors)

Most XAI models represent the explanations using attention maps (saliency). However, these explanations are difficult for humans to understand. For example, authors in [JW19] considered NLP tasks (text classification, natural language inference (NLI), and question answering) to show that attention mechanism is not useful for humans. Therefore, there is a dire need to represent and generate human-friendly explanations. Recent work by [KWG18] presents a first step towards this goal. They propose a technique called TCAV that takes the the user defined concept (X) represented using a set of example images and maps it to the activation space of any given layer l in the network. It then constructs a vector representation of each concept, called CAV (denoted as v_X), by using a direction normal to a linear classifier trained to distinguish between the concept activations from the random activations. The sensitivity of network predictions towards a concept is gauged by computing directional derivatives ($S_{c,X}$) to produce estimates of how important the concept X was for a CNN's prediction of a target class c, e.g. how important is the concept `stripedness` for predicting the zebra class.

$$S_{c,X} = \nabla g_c(f(I)) \cdot v_X \tag{2.1}$$

where g_c denote classifier component of CNN that takes output of f and predicts log-probability of output class c. Because TCAV provides explanations using high-level concepts, it is expected to achieve higher human trust and reliance values compared to the attention based explanations [SCD17a, RSG16].

2.2.4 Proxy or Surrogate Models

A Proxy or surrogate model is a simpler interpretable model that approximates the behaviour of the complex performer [RSG16, AJ18, ST01, AK12]. It reduces the complexity of the original performer but produces similar output estimates. Most surrogate XAI models are model-agnostic. A surrogate model that is trained to explain individual instances is referred to as local surrogate model. For example, LIME [RSG16] approximates performer with a local linear model that serves as a surrogate for the performer in the neighborhood of the input. Similarly, authors in [ST01, ZCN17] locally approximate neural networks with decision trees. This notion of using proxy models is also referred to as Knowledge Distillation [HVD15, HK18, PPA18] and Rule Extraction [ZMJ16].

2.2.5 Feature visualization

Feature visualization techniques typically identify qualitative interpretations of features used for making predictions or decisions. Recently, there has been an increased interest in developing feature visualizations for deep learning models, especially for Convolutional Neural Nets (CNNs) in computer vision applications, and Recurrent Neural Nets (RNNs) in NLP applications. For example, gradient ascent optimization is used in the image space to visualize the hidden feature layers of unsupervised deep architectures [EBC09]. Also, convolutional layers are visualized by reconstructing the input of each layer from its output [ZF14]. Recent visual explanation models seek to jointly classify the image and explain why the predicted class label is appropriate for the image [HAR16]. Other related work includes a visualization-based explanation framework for Naive Bayes classifiers [GPL03], an interpretable character-level language models for analyzing the predictions in RNNs [KJF15], and an interactive visualization for facilitating analysis of RNN hidden states [SGH16].

2.2.6 Perturbation Analysis

Perturbation analysis helps in measuring the feature importance for the predictions made by performer [FRD18, MFF17]. The assumption here is that performer's confidence in the prediction will be low if an important feature has been removed (or masked) after perturbing the input features. Adversarial analysis [GSS14] and Probing techniques [CKL19] are few popular techniques for perturbation analysis.

2.2.7 Counterfactual Explanations

Counterfactual (and Contrastive) explanations provide a *minimal* amount of information capable of altering a model's decision. In other words, they aim at describing the causal situations such as "What would be the output of model if X had not occurred?". This makes them easily digestible and practically useful for understanding the reasons for a model's decision [PGG18, WMR17, GWE19, VK19].

For example, [FV17] propose a counterfactual reasoning framework to find the part of an image most responsible for a classifier decision. This saliency based explanation framework helps in understanding where the model looks by discovering which parts of an image most affect its output score when perturbed. [GWE19] proposes a counterfactual explanation framework to identify how the input image could be changed such that the model would output a different specified class. To do this, they select a distractor image that the model predicts as class c_1 and identify spatial regions such that replacing the identified region in input image with the regions from distractor image would push the model towards classifying I as c_2. Contrastive explanations are proposed by [DCL18] to identify minimal and sufficient features to justify the classification result.

2.2.8 Partial Dependence Plots

Partial dependence plots (PD) is a model-agnostic XAI technique that helps in understanding the relationships between one or more input variables as well as marginal effect of a given variable on a performer's decision [Fri01, HTF01, Mol19].

2.2.9 Attribution

Attribution is a set of techniques that highlight pixels of the input image (saliency maps) that most caused the output classification. The following Gradient-based visualization methods [ZKL16, SCD17b] have been proposed to extract image regions responsible for the network output.

1. *Class Activation Mapping (CAM)* [ZKL16]. CAM produces attention map as explanation, i.e. it highlights the important regions in the image for predicting a target output.

2. *Gradient-weighted Class Activation Mapping (Grad-CAM)* [SCD17a]. Grad-CAM uses the gradients of target class flowing into the final convolutional layer to produce attention map as explanation.

3. *Layer-wise Relevance Propagation (LRP)* [BBM15]. LRP generates attention map by propagating classification probability backward through the network and then calculates relevance scores for all pixels.

4. SmoothGrad [STK17]. Smooth grad produces attention map as explanation by adding gaussian noise to the original image and then calculating gradients multiple times and averaging the results.

More recently, in addition to the above techniques, other important lines of research in explainable AI explore dimensionality reduction techniques [Bri, MH08]. Also, influence measures [DDP15] have been used to identify the importance of features in affecting the classification outcome for individual data points. There are several works [Mil18a, Hil90, Lom06] on the goodness measures of explanation which aim to understand the underlying characteristics of explanations.

2.3 X-ToM Framework

Our X-ToM consists of three main components:

- A Performer that generates image interpretations (i.e., machine's mind represented as pg^M) using a set of computer vision algorithms;
- An Explainer that generates maximum utility explanations in a dialog with the user by accounting for pg^M and pg^{UinM} using reinforcement learning;
- An Evaluator that quantitatively evaluates the effect of explanations on the human's understanding of the machine's behaviors (i.e., pg^{MinU}) and measures human trust by comparing pg^{MinU} and pg^M.

2.3.1 X-ToM Collaborative Task

As part of our X-ToM framework, we have designed a collaborative task-solving game for visual recognition. The game consists of two phases. In the first phase, the user is shown a blurred image and given a task to recognize what the image shows. X-ToM has access to the original (unblurred) image and the machine's (i.e. **Performer's**) inference result pg^M (see Section 2.3.3). The user is allowed to ask questions regarding objects and parts in the image that the user finds relevant for his/her own recognition task. Using the detected objects and parts in pg^M, X-ToM **Explainer** provides visual explanations to the user, as shown in Figure 2.3. This process allows the machine to infer what the user sees and iteratively update pg^{UinM}, and thus select an optimal explanation at every turn of the game (see Section 2.3.4). Optimal explanations generated by the **Explainer** are the key to maximize the human trust in the machine. The second phase is specifically designed for evaluating whether the explanation provided in the first phase helps the user understand the system behaviors. The **Evaluator** shows a set of original (unblurred) images to the user that are similar to (but different from) the ones used in the first phase of the game (i.e., the set of images shows the same class of objects or human activity). The user is then given a task to predict in each image the locations of objects and parts that would be detected by the machine (i.e., in pg^{M}) according to his/her understanding of the machine's behaviors. Based on the human predictions, the **Evaluator** estimates pg^{MinU} and quantifies human trust in the machine by comparing pg^{MinU} and pg^{M} (see Section 2.3.5).

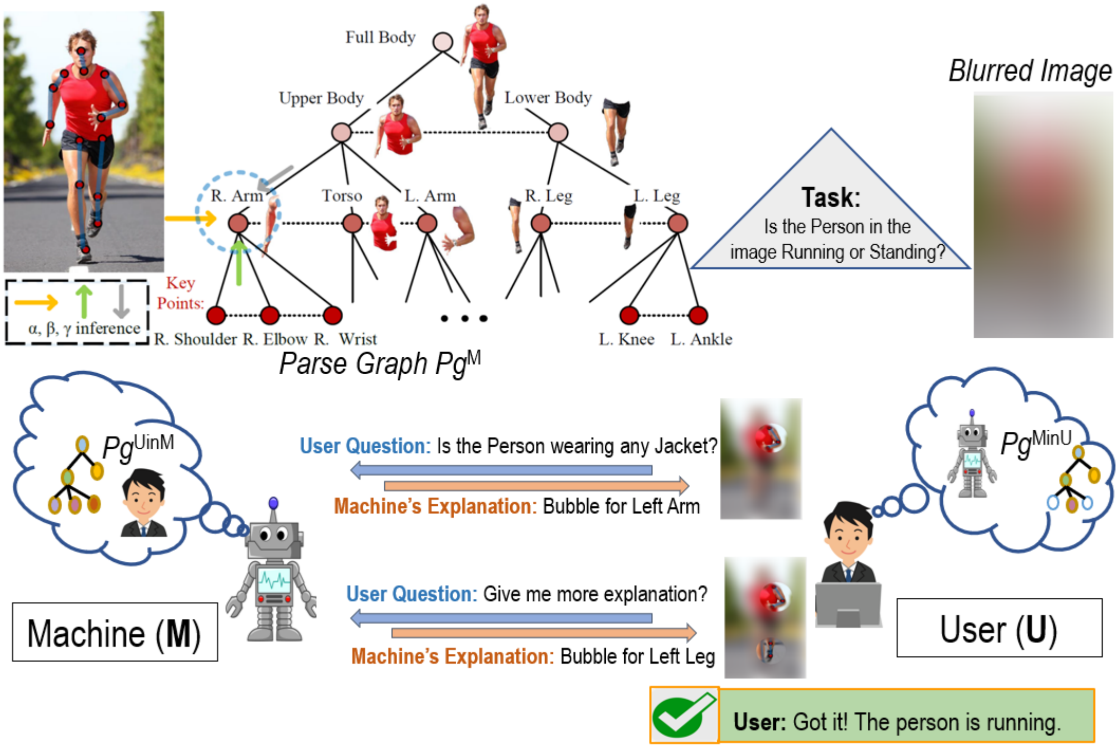

Full Body
Upper Body
Lower Body
R. Arm
Torso
L. Arm
R. Leg
L. Leg
Key Points:
R. Shoulder R. Elbow R. Wrist
L. Knee L. Ankle
α, β, γ inference
Parse Graph Pg^M
Task:
Is the Person in the image Running or Standing?
Blurred Image
Pg^UinM
Pg^MinU
User Question: Is the Person wearing any Jacket?
Machine's Explanation: Bubble for Left Arm
User Question: Give me more explanation?
Machine's Explanation: Bubble for Left Leg
Machine (M)
User (U)
User: Got it! The person is running.

Figure 2.3: An example of the first phase of an X-ToM game aimed at estimating pg^{UinM}: The user is shown a blurred image and given a task to recognize if the person in the image is running or walking. X-ToM has access to the original (unblurred) image and pg^{M}. The user then asks questions regarding objects and parts in the image. Using the detections in pg^{M}, X-ToM provides visual explanations as "bubbles" that reveal the corresponding image parts in the blurred image. The generated explanations are used to update pg^{UinM}.

2.3.2 Representation of Minds in X-ToM

The three minds pg^{M}, pg^{MinU}, and pg^{UinM} are sub-graphs of an And-Or Graph (AOG) defining all objects, parts, and their relationships and attributes of the visual domain considered. Our motivation to use AOGs for modeling the three mental states of the Theory of Mind stems from the following advantages. First, an AOG is a context-sensitive stochastic grammar [ZM07] that can explicitly capture rich contextual and hierarchical relationships (spatial, temporal and causal). Second, AOG based representation and inference is a domain generic approach and the literature has abundantly demonstrated that AOG based systems, especially recent methods that combine deep learning and AOGs, are the top performers for a wide range of tasks in domains such as computer vision, natural language processing, and human-robot collaboration [ZWZ18, LYS16, WZ11, TML14, PNZ18]. Third, since the result of visual recognition (i.e., a parse graph) is a sub-graph of the AOG, image interpretations can be readily explained using the top-down, bottom-up, or contextual types of visual reasoning enabled by the AOG. Finally, and of great importance for XAI systems, the rich contextual and hierarchical nature of AOGs allows for formalizing and quantitatively evaluating human trust in the visual performer along both depth and breadth.

As AOG is interpretable, why not show Pg^M directly to the user as an explanation? It will be daunting to show the entire AOG since our AOG encodes hundreds of objects, parts, activities, attributes and other concepts as nodes. In addition, AOG has numerous edges. It might be possible to visualize a part of AOG, but it is not clear how to optimize which AOG subgraph would not overwhelm the user and maximize utility. The advantage of using our dialog based explanations is that, at each dialog turn, explainer can tailor the explanations based on the user's current perception and understanding [Mil18a].

2.3.3 X-ToM Performer (for Image Interpretation)

In this Chapter, the visual tasks involve detecting and localizing human body parts, identifying their poses and attributes, and recognizing human actions from a given image. The AOG for this visual domain uses AND nodes to represent decompositions of human body parts into subparts, and OR nodes for alternative decompositions. Each node is characterized by attributes that pertain to the corresponding human body part, including the pose and action of the entire body. Also, edges in the AOG capture hierarchical and contextual relationships of the human body parts.

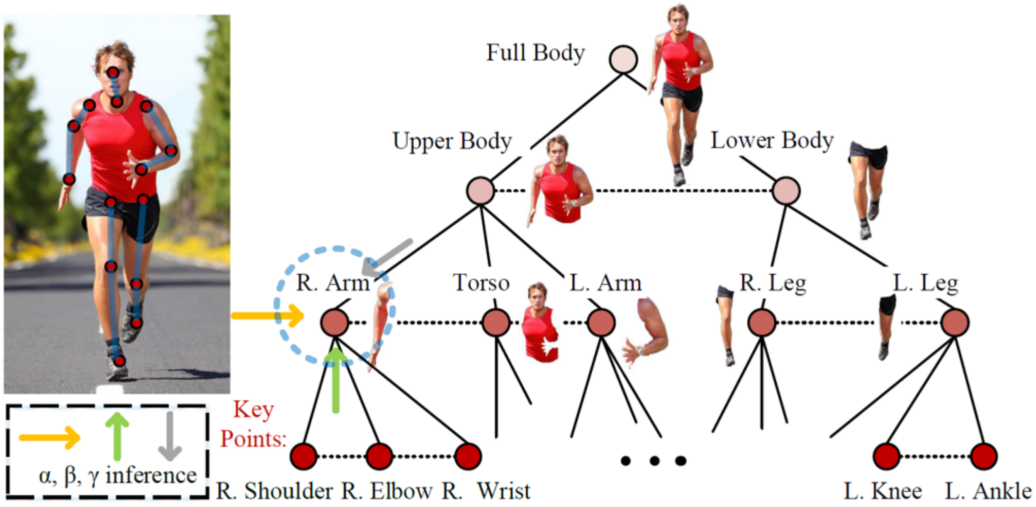


Figure 2.4: Illustration of the α,β,γ inference processes in the AOG for human body pose detection

Our AOG-based performer uses three inference processes α, β and γ at each node. Figure 2.3 shows an example part of the AOG relevant for human body pose estimation [PNZ18]. The α process detects nodes (i.e., human body parts) of the AOG directly based on image features, without taking advantage of the surrounding context. The β process infers nodes of the AOG by binding the previously detected children nodes in a bottom-up fashion, where the children nodes have been detected by the α process (e.g., detecting human's upper body from the detected right arm, torso, and left arm). Note that the β process is robust to partial object occlusions as it can infer an object from its detected parts. The γ process infers a node of the AOG top-down from its previously detected parent nodes, where the parents have been detected by the α process (e.g., detecting human's right leg from the detected outline of the lower body). The parent node passes contextual information so that the performer can detect the presence of an object or part from its surround. Note that the γ process is robust to variations in scale at which objects appear in images.

2.3.4 X-ToM Explainer (for Explanation Generation)

The explainer, in the first phase of the game, makes the underlying α, β, and γ inference process of the performer more transparent to the human through a collaborative dialog. At one end, the explainer is provided access to an image and the performer's inference result pg^M on that image. At the other end, the human is presented a blurred version of the same image, and asked to recognize a body part, or pose, or human action depicted (e.g., whether the person is running or walking). To solve the task, the human may ask the explainer various "what", "where" and "how" questions (e.g., "Where is the left arm in the image"). We make the assumption that the human will always ask questions that are related to the task at hand so as to solve it efficiently. The explainer answers these questions using pg^M and justifies the answers by showing the corresponding visual explanations in the image (as illustrated in Figure 2.5).

As visual explanations, we use "bubbles" [GS01], where each bubble reveals a circular part of the blurred image to the human. The bubbles coincide with relevant image parts for answering the question from the human, as inferred by the performer in pg^M. For example, a bubble may unblur the person's left leg in the blurred image, since that image part has been estimated in pg^M as relevant for recognizing the human action "running" occurring in the image.

Following the "principle of least collaborative effort" [CW86] and the aforementioned findings [Mil18a] that explanations should *not* overwhelm the human, our X-ToM explainer utilizes pg^M and pg^{UinM} (i.e., the contextual and hierarchical relationships explicitly modeled in the AOG) for controlling the depth and breadth of explanations. To enable this control, each bubble is characterized by a number of parameters, including the amount of image reveal (i.e., the unblurring level), size, and location in the image, to name a few. We use reinforcement learning to train the explainer to optimize these parameters and thus provide optimal visual explanations.

2.3.5 X-ToM Evaluator (for Trust Estimation)

The second phase of the X-ToM game serves to assess the effect of the explainer on the human's understanding of the performer. This assessment is conducted by the evaluator. The human is presented with a set of (unblurred) images that are different from those used in the first phase. For every image, the evaluator asks the human to predict the performer's output. The evaluator poses multiple-choice questions and the user clicks on one or more answers. As shown in Figure 2.6, we design these questions to capture different aspects of human's understanding of α, β and γ inference processes in the performer. Based on responses from the human, the evaluator estimates pg^{MinU}. By comparing pg^{MinU} with the actual machine's mind pg^{M} (generated by the performer), we have defined the following qualitative and quantitative metrics to quantitatively assess human trust [Hof17b, HHB10, HMK18, Mil18b] in the performer:

Quantitative Metrics:

(1) *Justified Positive and Negative Trust:* It is possible for humans to feel positive trust with respect to certain tasks, while feeling negative trust (i.e. mistrust) on some other tasks. The positive and negative trust can be a mixture of justified and unjustified trust [Hof17b, HMK18]. We compute justified positive trust (JPT) and negative trust (JNT) as follows:

$$\begin{aligned}
\text{JPT} &= \frac{1}{N}\sum_{i}\sum_{z=\alpha,\beta,\gamma} \Delta\text{JPT}(i,z),\\
\Delta\text{JPT}(i,z) &= \frac{\|pg^{MinU}_{i,z,+} \cap pg^{M}_{i,+}\|}{\|pg^{M}_{i,+}\|},\\
\text{JNT} &= \frac{1}{N}\sum_{i}\sum_{z=\alpha,\beta,\gamma} \Delta\text{JNT}(i,z),\\
\Delta\text{JNT}(i,z) &= \frac{\|pg^{MinU}_{i,z,-} \cap pg^{M}_{i,-}\|}{\|pg^{M}_{i,-}\|},
\end{aligned}$$

where N is the total number of games played. z is the type of inference process. $\Delta\text{JPT}(i,z)$, $\Delta\text{JNT}(i,z)$ denote the justified positive and negative trust gained in the i-th turn of a game on the z inference process respectively. $pg^{MinU}_{i,z,+}$ denotes nodes in pg^{MinU}_{i} for which the user thinks the performer is able to accurately detect in the image using the z inference process. Similarly, $pg_{i,z,-}^{MinU}$ denotes nodes in pg_i^{MinU} for which the user thinks the performer would fail to detect in the image using the z inference process. $\|pg\|$ is the size of pg. Symbol $\cap$ denote the graph intersection of all nodes and edges from two pg's.

(2) *Reliance:* Reliance (Rc) captures the extent to which a human can accurately predict the performer's inference results without over- or under-estimation. In other words, Reliance is proportional to the sum of JPT and JNT.

$$\text{Rc} = \frac{1}{N} \sum_{i} \sum_{z=\alpha,\beta,\gamma} \Delta\text{Rc}(i,z),$$
$$\Delta\text{Rc}(i,z) = \frac{\|pg_{i,z}^{MinU} \cap pg_{i,z}^{M}\|}{\|pg_i^M\|}.$$

Qualitative Metrics:
(3) *Explanation Satisfaction (ES)*. We measure users' feeling of satisfaction at having achieved an understanding of the machine in terms of usefulness, sufficiency, appropriated detail, confidence, accuracy, and consistency. We ask them to rate each of these metrics on a Likert scale of 0 to 9.

2.4 Experiments

We deployed the X-ToM game on the Amazon Mechanical Turk (AMT) and trained the X-ToM Explainer through the interactions with turkers. All the turkers have a bachelor's degree or higher. We used three visual recognition tasks in our experiments, namely, human body parts identification, pose estimation, and action identification. We used 1000 images randomly selected from Extended Leeds Sports (LSP) dataset [JE10]. Each image is used in all the three tasks. During training, each trial consists of one X-ToM game where a turker solves a given task on a given image. We restrict Turkers from solving a task on an image more than once. In total, about 2400 unique workers contributed in our experiments.

We performed off-policy updates after every 200 trials, using Adam optimizer [KB15] with a learning rate of 0.001 and gradients were clipped at [-5.0, 5.0] to avoid explosion. We used ϵ-greedy policy, which was annealed from 0.6 to 0.0. We stopped the training once the model converged. In our case, the X-ToM policy model converged after interacting with 3500 turkers. All our data and code will be made publicly available.

Elaboration	Sequence	Recurrence	Restatement	Summary
26%	48.7%	12.6%	5.1%	7.6%

Table 2.1: Distribution of observed discourse relations in the test trials

The trained X-ToM Explainer was applied to an additional 500 X-ToM games with AMT turkers for testing. Table 2.1 shows the percentage of discourse relations among bubbles found in the test interactions. As can be seen, the discourse relation `sequence` dominates other relations. This indicates that the X-ToM's most common explanation strategy is to prefer a bubble containing new evidence (that was not already shown to the user). Furthermore, the experiment has shown that 55.3% of the bubbles in the test trials were generated using α explanation act, 23.1% using β explanation act, and 21.6% using γ explanation act. The high percentage of β and γ explanation acts indicate that contextual evidence is not only helpful for the performer to detect but also for the explainer to explain.

2.4.1 AMT Evaluation of X-ToM Explainer

We conducted an ablation study to quantify the importance of taking the inferred human's mind into account for generating optimal explanations, i.e., the ablated model does not explicitly represent and infer pg^{UinM}. Similar to X-ToM, the ablated model was also deployed and trained on AMT. The trained ablated model was again applied to an additional 500 X-ToM games with AMT turkers for testing. Table 2.2 compares X-ToM Explainer with the ablated model in terms of objective measures such as average success rate (ss), average number of bubbles, average rewards (r). X-ToM Explainer significantly outperforms the ablated model ($p < 0.01$) in terms of the overall reward. Although the success rates of both models are similar, the ablated model is found to use a significantly larger number of bubbles, which leads to lower overall reward.

Model	#test trials	ss	#bubbles	r
X-ToM	500	**81.3%**	**10.5**	**0.91**
Ablated Model	500	77.1%	28	0.42
Human Strategy	100	78.9%	**6**	0.62

Table 2.2: Comparison of X-ToM with ablated and human baselines

Figure 2.7 compares the justified positive trust (JPT), justified negative trust (JPT), and Reliance (Rc) of X-ToM with the baselines.

Using an additional 100 X-ToM games on AMT, we further compare the explanations generated by our X-ToM Explainer with the explanations annotated by humans. We asked three graduate students (not the authors), to select the most appropriate bubbles for a given task. Bubbles that have been agreed upon by these three subjects were taken as the best explanations for the given task and image. In terms of maximizing the reward, we found that X-ToM Explainer performed significantly better than the human strategy of bubble selection ($p < 0.01$). However, we found that the average dialog length in the human explanations is 6, while the average dialogue length observed in the X-ToM explanations is 10.5, indicating that there is a possibility to further improve the quality of the X-ToM explanations. We leave this for future exploration.

Using X-ToM Evaluator, we conduct human subject experiments to assess the effectiveness of the X-ToM Explainer, that is trained on AMT, in increasing human trust through explanations. We recruited 120 human subjects from our institution's Psychology subject pool [*]. These subjects have

[*] These experiments were reviewed and approved by our institution's IRB.

no background on computer vision, deep learning and NLP. We applied between-subject design and randomly assigned each subject into one of the three groups. One group used X-ToM Explainer, and two groups used the following two baselines respectively:

- Ω_{QA}: we measure the gains in human trust only by revealing the answers for the tasks without providing any explanations to the human.
- Ω_{Salience}: in addition to the answers, we also provide saliency maps generated using attribution techniques to the human as explanations [ZKL16, SCD17b].

Within each group, each subject will first go through an introduction phase where we introduce the tasks to the subjects. Next, they will go through familiarization phase where the subjects become familiar with the machine's underlying inference process (Performer), followed by a testing phase where we apply our trust metrics and assess their trust in the underlying Performer.

As we can see, JPT, JNT and Rc values of X-ToM are significantly higher than Ω_{QA} and Ω_{Salience} ($p < 0.01$). *Also, it should be noted that attribution techniques ($\Omega_{Salience}$) did not perform any better than the Ω_{QA} baseline where no explanations are provided to the user.* This could be attributed to the fact that, though saliency maps help human subjects in localizing the region in the image based on which the performer made a decision, they do not necessarily reflect the underlying inference mechanism. In contrast, X-ToM Explainer makes the underlying inference processes (α, β, γ) more explicit and transparent and also provides explanations tailored for individual user's perception and understanding. Therefore X-ToM leads to the significantly higher values of JPT, JNT and Rc. This is one of the key results of our work, given the popularity of attribution techniques as the state-of-the-art explanations.

Figure 2.8 shows the average explanation satisfaction rates obtained from each of the three groups. As we can see, subjects in X-ToM experiment group found that explanations were highly useful, sufficient and detailed compared to the baselines ($p < 0.01$). Interestingly, we did not find significant differences across the three groups in terms of other satisfaction measures: confidence, understandability, accuracy and consistency. We leave this observation for future exploration.

2.4.2 Gain in Reliance over time

We hypothesized that human trust and reliance in machine might improve over time. This is because, it can be harder for humans to fully understand the machine's underlying inference process in one single session. Therefore, we conduct an additional experiment with eight human subjects where the subjects' reliance is measured after every session. The results are shown in Figure 2.9. As we expected, subjects' reliance increased over time. Specifically, reliance with respect to α inference process significantly improved only after 2.5 sessions. Reliance with respect to β and γ inference processes significantly improved after 4.5 sessions. It is clearly evident that, with more sessions, it is possible to further improve human reliance in AI system.

2.4.3 Case Study

Figure 2.10 shows examples where the top-3 best explanations preferred by X-ToM are compared against the top-3 explanations generated by the attribution techniques. The first column shows the input image for the task. The second column shows all the evidence (i.e., explanations in the form of bubbles, highlighted in yellow color) used in the machine's inference about the task. The thicker the bubble, the higher is its influence, for the machine, in interpreting the image. As we can see, attribution techniques chose the explanations only based on how influential they are for the machine in recognizing the image (third column). In contrast, since X-ToM maximizes the utility of explanations based on both influence values and user's model, explanations selected by the X-ToM (fourth column) are diverse and are more intuitive for humans to understand and solve the task efficiently. For example, for the first image, to aid the human user in solving the task 'Is the person in the image walking', X-ToM generates the explanation bubbles based on left arm, right arm and lower body of the person, whereas attribution techniques generate the top-3 bubbles only based on right arm which clearly is not sufficient for the user to successfully solve the task.

In addition to the quantitative and qualitative metrics discussed in the previous section, we also measure the following metrics for comparing our X-ToM framework with the baselines:

- **Response Time**: We record the time taken by the human subject in answering evaluator questions. Figure 2.14 shows the average response times (in milliseconds per question) for each of the three groups (X-ToM, QA and Saliency Maps). We expected the participants in X-ToM group to take less time to respond compared to the baselines. However, we find no significant difference in the response times across the three groups.

- **Subjective Evaluation of Reliance**: We collect subjective Reliance values (on a Likert scale of 0 to 9) from the subjects in the three groups. The results are shown in Figure 3.3. These results are consistent with our quantitative reliance measures. It may be noted that subjects' qualitative reliance in Saliency Maps is lower compared to the QA baseline.

2.5 Summary

This Chapter demonstrated X-ToM – a new framework for Explainable AI (XAI) and human trust evaluation based on the Theory-of-Mind (ToM). X-ToM generates explanations in a dialog by explicitly modeling, learning, and inferring three mental states based on And-Or Graphs – namely, machine's mind, human's mind as inferred by the machine, and machine's mind as inferred by the human. This allows for a principled formulation of human trust in the machine. For the task of visual recognition, we proposed a novel, collaborative task-solving game that can be used for collecting training data and thus learning the three mental states, as well as a testbed for quantitative evaluation of explainable vision systems. We demonstrated the superiority of X-ToM in gaining human trust relative to baselines.

2.6 Appendix

2.6.1 X-ToM Evaluator Interface and Questions

Specifically, there are two main types of evaluator questions about the user's prediction: (1) whether the Performer would successfully or incorrectly detect objects, parts and other concepts encoded by AOG; and (2) which image parts are most influential for the Performer's successful or incorrect object detection. For example, the evaluator's questions include "which parts of the image are most important for the machine to recognize that the person is running", and "which small part of image contributes most to inferring the surrounding larger part of image". Figures 2.16 to 2.18 show few sample screenshots (from our web interface) of the exact questions, on the detection of the body part "Left-Arm", that we pose to the subjects.

2.6.2 Evaluation with Psychology Subject Pool

Figure 2.13 shows the statistics (Age, First Language, Gender) of the 120 human subjects, recruited from our institution's Psychology subject pool.

2.6.3 Human Subject Evaluation: Additional Results

In addition to the metrics Justified Trust and Reliance, we also measure the following metrics for comparing our X-ToM framework with the baselines (QA and Saliency Maps):

- **Response Time**: We record the time taken by the human subject in answering evaluator questions. Figure 2.14 shows the average response times (in milliseconds per question) for each of the three groups (X-ToM, QA and Saliency Maps). We expected the participants in X-ToM group to take less time to respond compared to the baselines. However, we find no significant difference in the response times across the three groups.

- **Explanation Satisfaction**: We measure human subjects' feeling of satisfaction at having achieved an understanding of the machine in terms of usefulness, sufficiency, appropriated detail, confidence, understandability, accuracy and consistency [Hof17b, HMK18, Mil18b, HHB10]. We ask them to rate each of these metrics on a Likert scale of 0 to 9. Figure 2.8 shows the average explanation satisfaction rates obtained from each of the three groups. As we can see, subjects in X-ToM experiment group found that explanations were highly useful, sufficient and detailed compared to the baselines ($p < 0.01$). However, we did not find significant differences across the three groups in terms of other satisfaction measures: confidence, understandability, accuracy and consistency.

- **Subjective Evaluation of Reliance**: We collect subjective Reliance values (on a Likert scale of 0 to 9) from the subjects in the three groups. The results are shown in Figure 3.3. These results are consistent with our quantitative reliance measures. It may be noted that subjects' qualitative reliance in Saliency Maps is lower compared to the QA baseline.

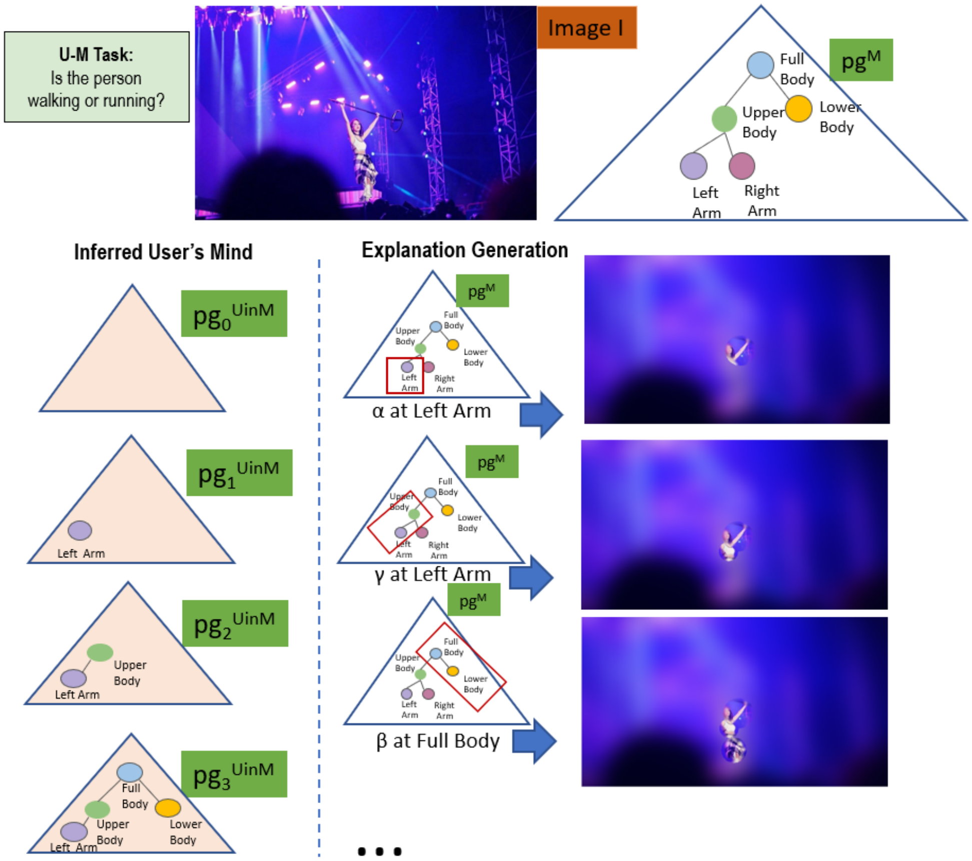
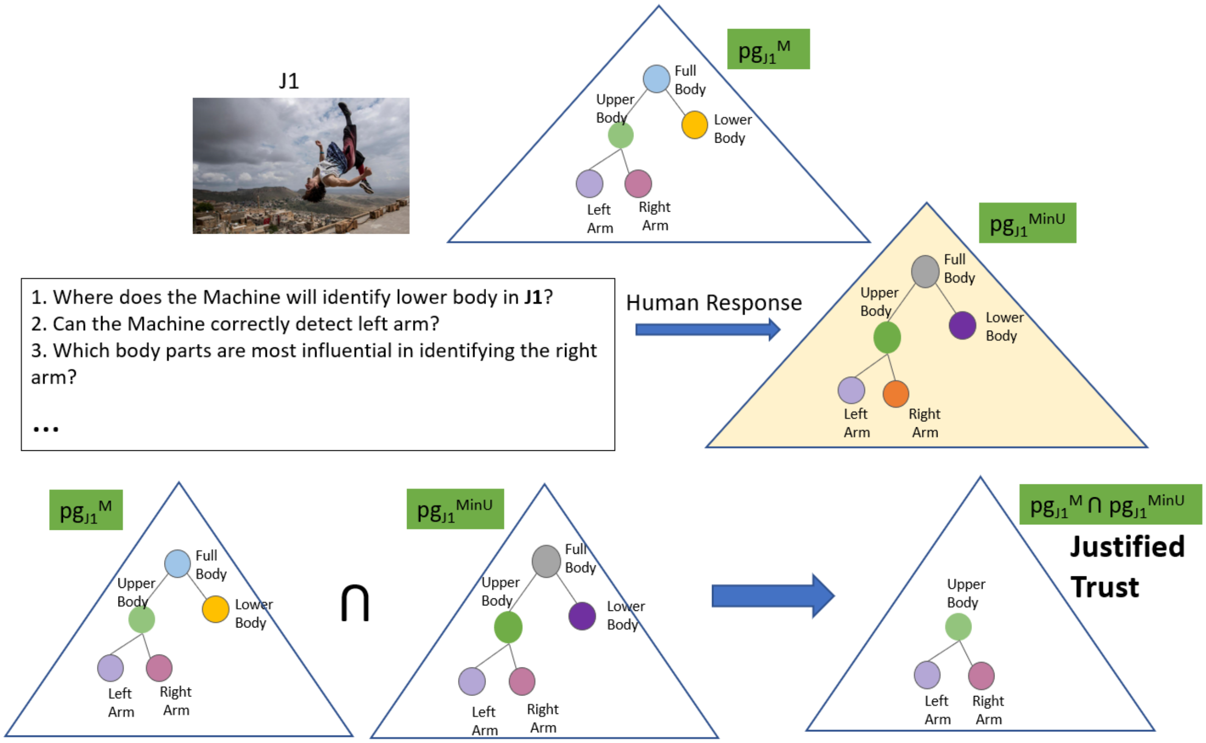


Figure 2.5: Illustration of the first phase in X-ToM game. The human is asked to solve the task "Is the person in the image walking or running?". The human may ask questions related to body parts and body poses. The machine reveals a bubble (of various sizes and scales) for each of those questions. The figure shows examples of explanations generated using α, β and γ processes and the updated inferred user's mind after each explanation.



1. Where does the Machine will identify lower body in **J1**?
2. Can the Machine correctly detect left arm?
3. Which body parts are most influential in identifying the right arm?

...

Figure 2.6: An example of second phase of X-ToM game where we estimate pg^{MinU} and also quantitatively compute justified trust.

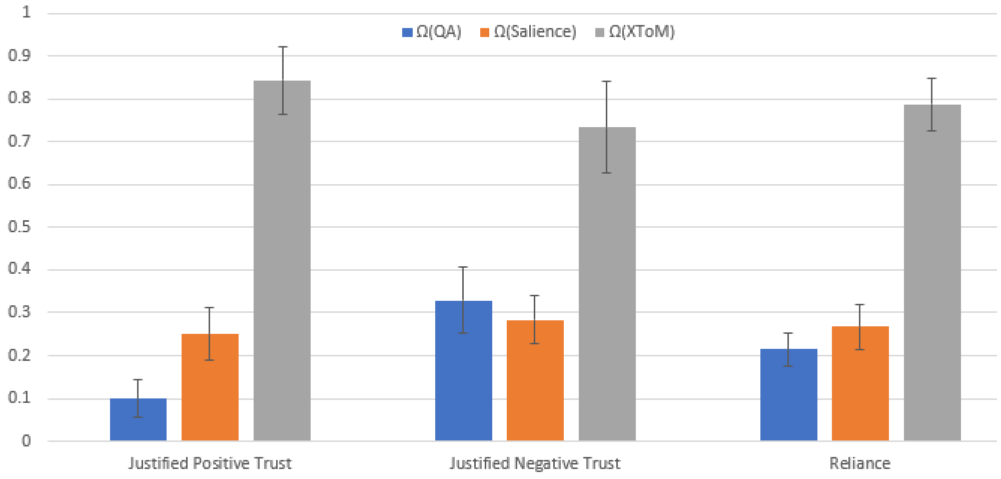


Figure 2.7: Gain in Justified Positive Trust, Justified Negative Trust and Reliance: X-ToM vs baselines (QA, Saliency Maps). Error bars denote standard errors of the means.

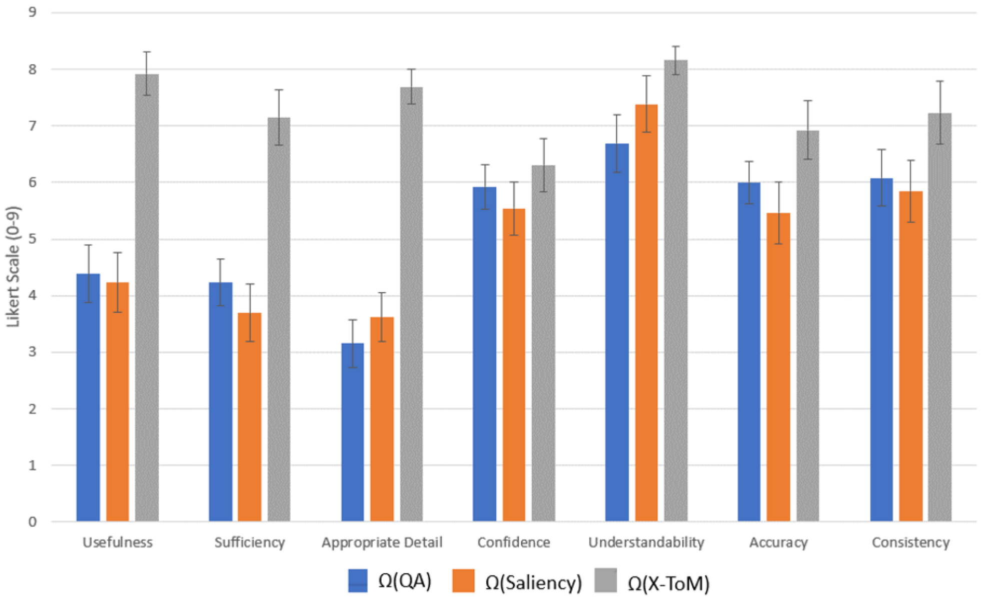


Figure 2.8: Explanation Satisfaction: X-ToM vs baselines (QA, Saliency Maps). Error bars denote standard errors of the means.

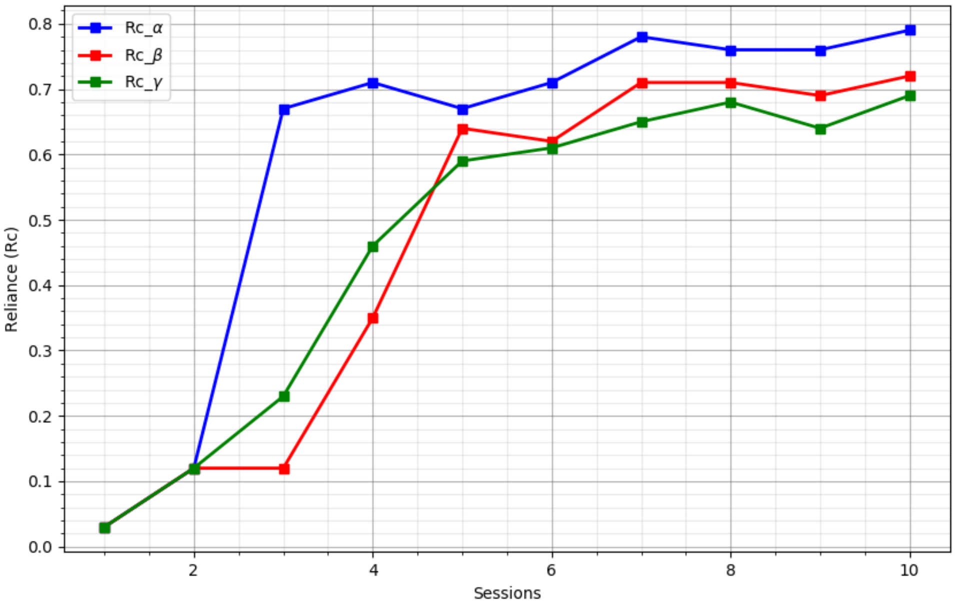


Figure 2.9: Gain in Reliance over sessions w.r.t α, β and γ processes

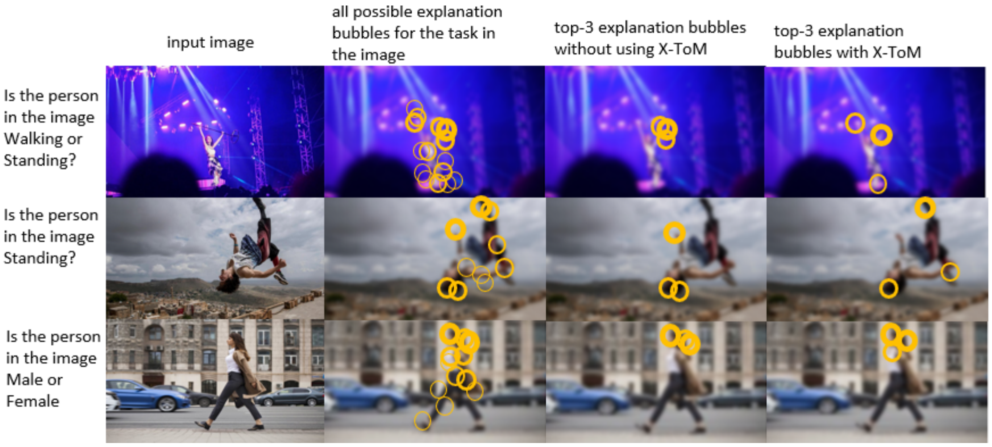

input image
all possible explanation bubbles for the task in the image
top-3 explanation bubbles without using X-ToM
top-3 explanation bubbles with X-ToM
Is the person in the image Walking or Standing?
Is the person in the image Standing?
Is the person in the image Male or Female

Figure 2.10: Top-3 best explanations generated with and without using X-ToM.

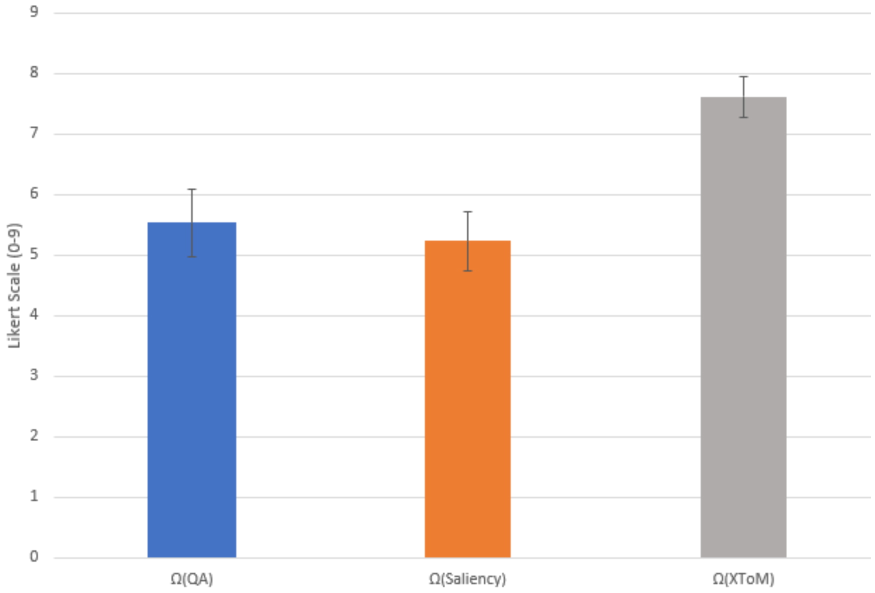


Figure 2.11: **Qualitative Reliance**. Error bars denote standard errors of the means.

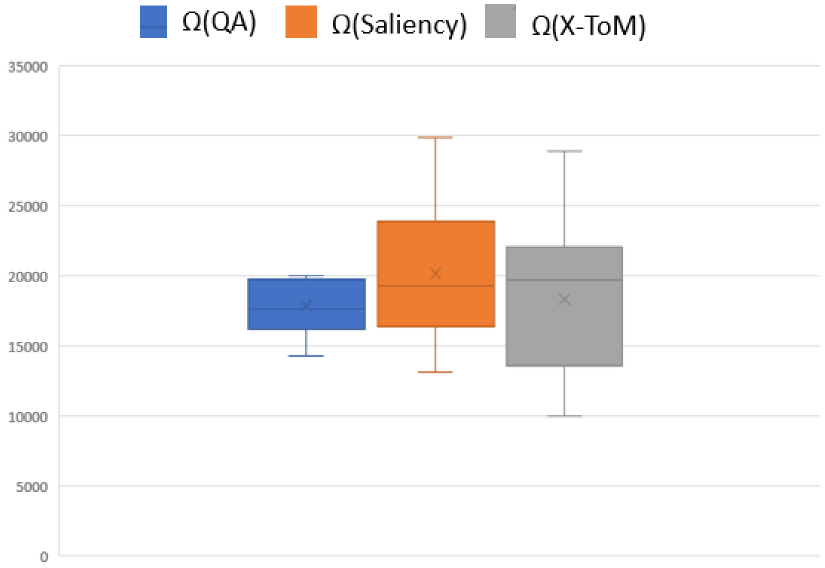


Figure 2.12: **Response Times** (in milliseconds per question). Error bars denote standard errors of the means.

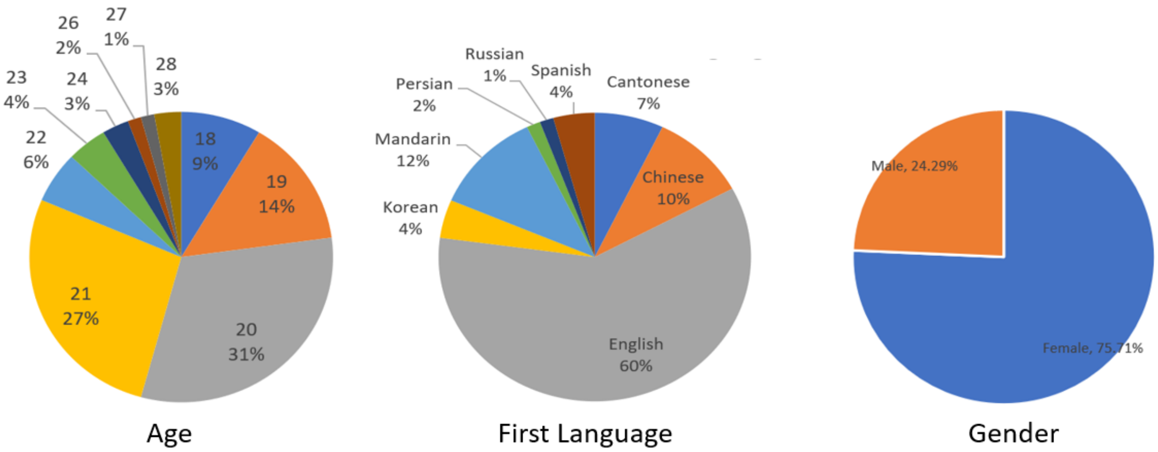


Figure 2.13: Statistics (based on Age, First Language and Gender) of the 120 human subjects, from Psychology subject pool, participated in our study.

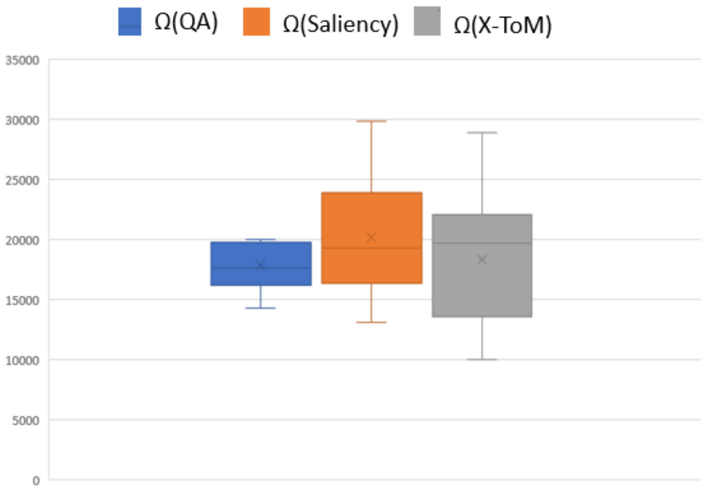


Figure 2.14: **Response Times** (in milliseconds per question). Error bars denote standard errors of the means.

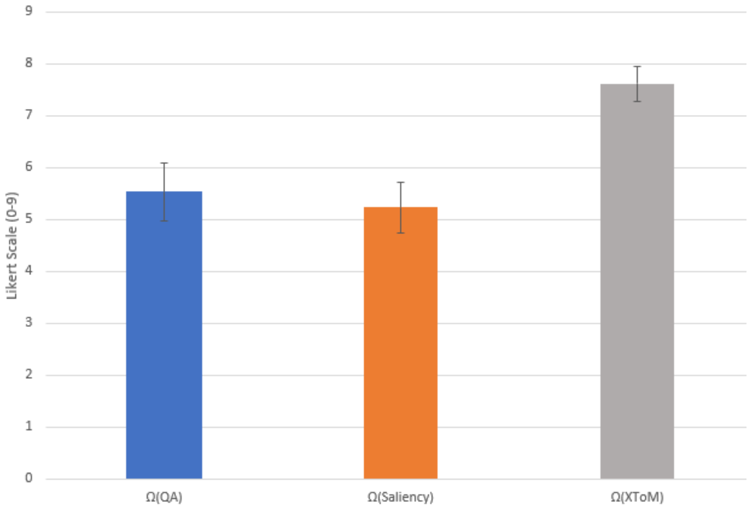
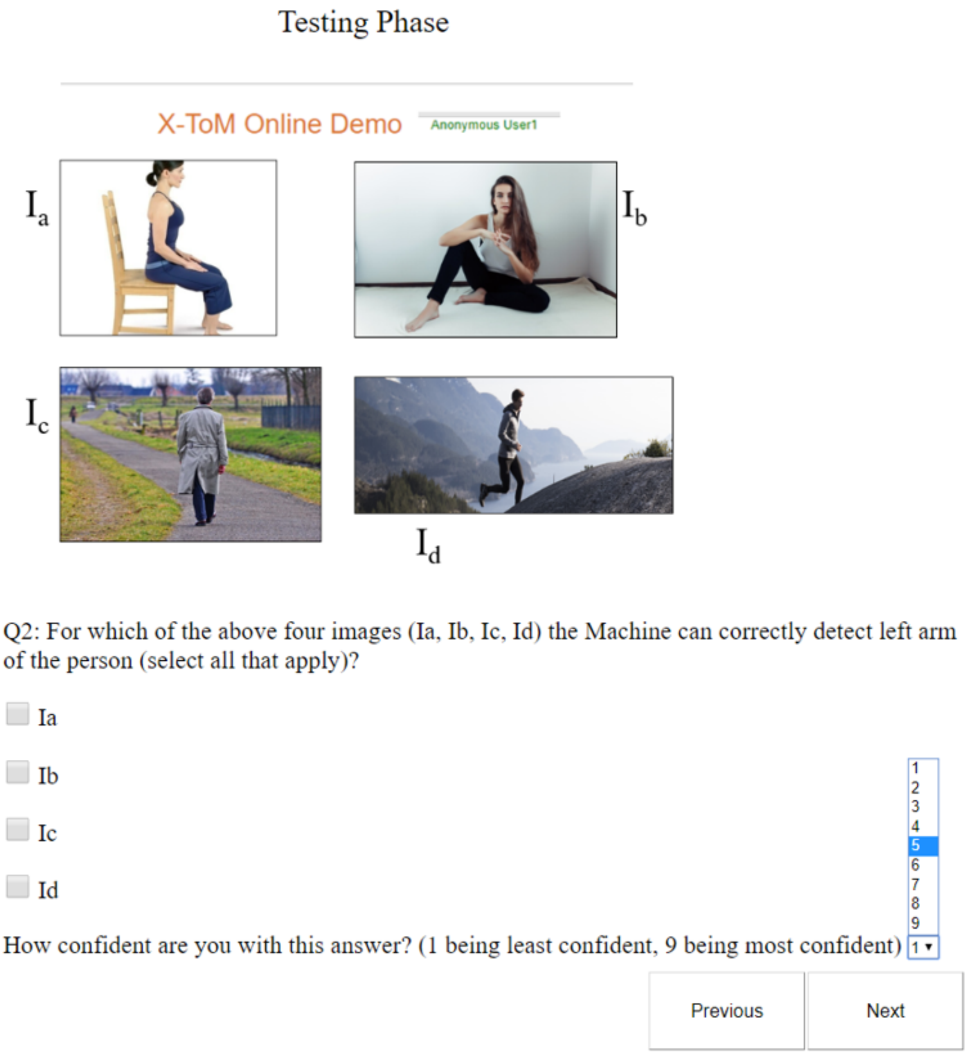


Figure 2.15: **Qualitative Reliance**. Error bars denote standard errors of the means.



Testing Phase

X-ToM Online Demo

Anonymous User1

Q2: For which of the above four images (Ia, Ib, Ic, Id) the Machine can correctly detect left arm of the person (select all that apply)?

□ Ia

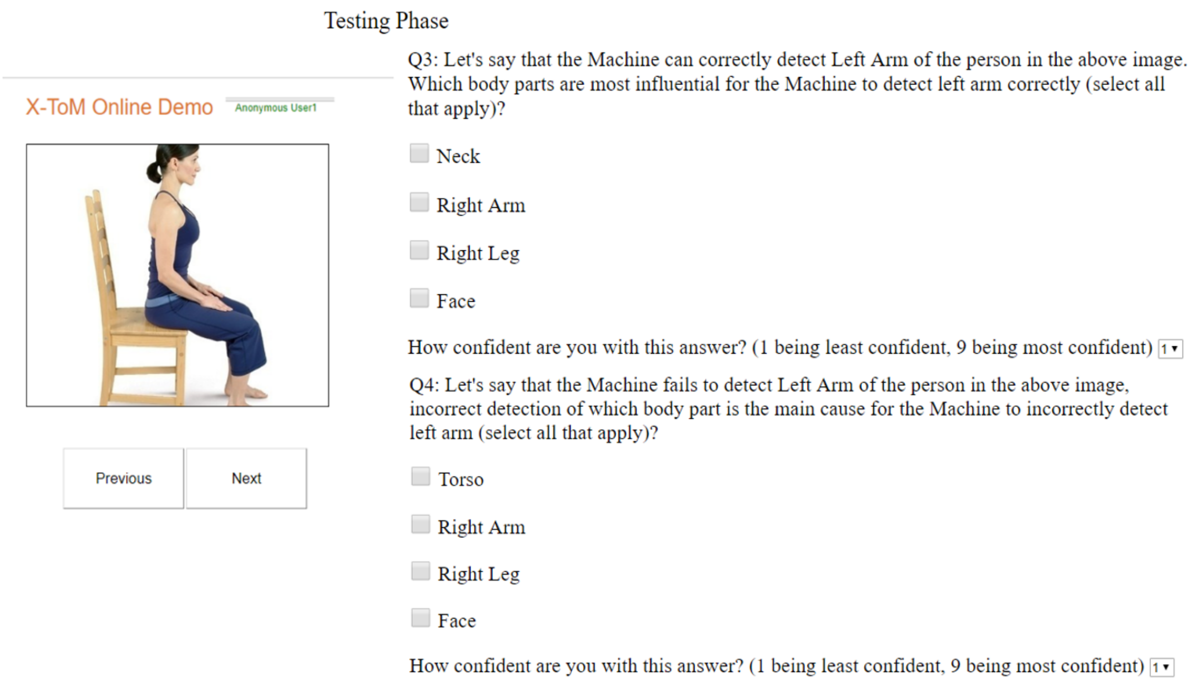
□ Ib

□ Ic

□ Id

How confident are you with this answer? (1 being least confident, 9 being most confident) 1 ▾

Figure 2.16: Sample evaluator questions



Testing Phase

Q3: Let's say that the Machine can correctly detect Left Arm of the person in the above image. Which body parts are most influential for the Machine to detect left arm correctly (select all that apply)?

□ Neck

□ Right Arm

□ Right Leg

□ Face

How confident are you with this answer? (1 being least confident, 9 being most confident) 1 ▾

Q4: Let's say that the Machine fails to detect Left Arm of the person in the above image, incorrect detection of which body part is the main cause for the Machine to incorrectly detect left arm (select all that apply)?

□ Torso

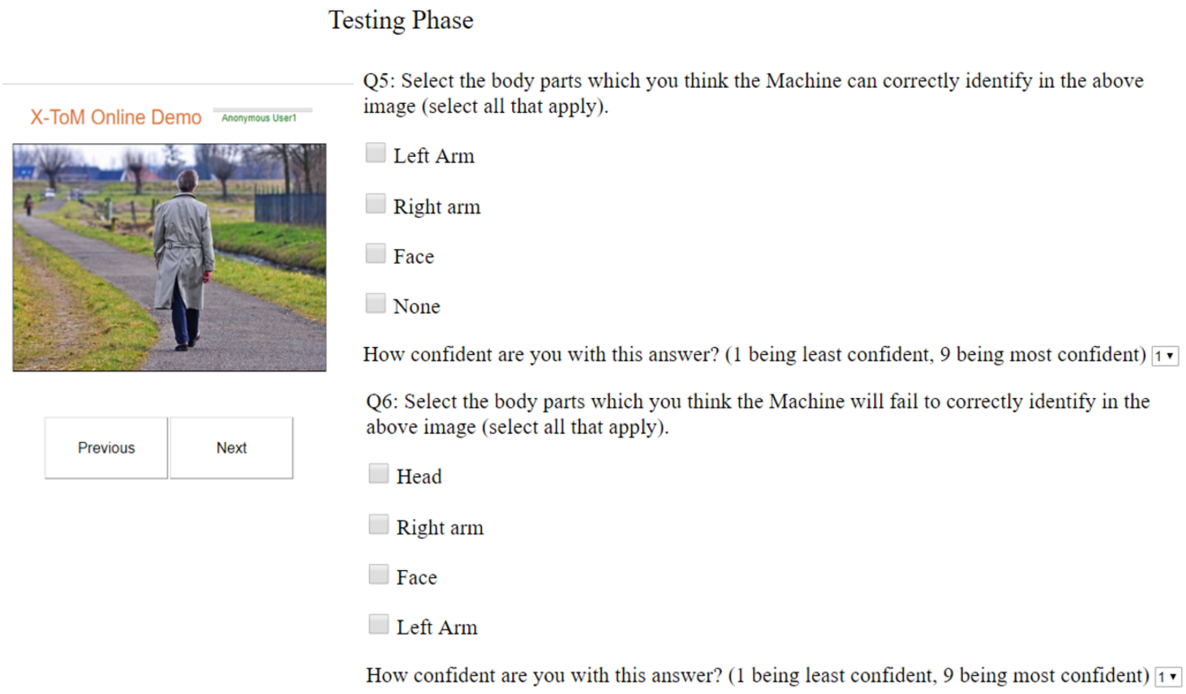
□ Right Arm

□ Right Leg

□ Face

How confident are you with this answer? (1 being least confident, 9 being most confident) 1 ▾

Figure 2.17: Sample evaluator questions



Testing Phase

Q5: Select the body parts which you think the Machine can correctly identify in the above image (select all that apply).

□ Left Arm

□ Right arm

□ Face

□ None

How confident are you with this answer? (1 being least confident, 9 being most confident) 1 ▾

Q6: Select the body parts which you think the Machine will fail to correctly identify in the above image (select all that apply).

□ Head

□ Right arm

□ Face

□ Left Arm

How confident are you with this answer? (1 being least confident, 9 being most confident) 1 ▾

Figure 2.18: Sample evaluator questions

CHAPTER 3

Conceptual and Counterfactual Explanations

The previous Chapter introduced an iterative and collaborative explanation framework X-ToM by explicitly modeling, learning, and inferring three mental states based on And-Or Graphs. While the explanation bubbles in X-ToM reveal the optimal explanation path, they cannot capture high-level semantics/concepts of the features or attributes in the dataset without explicitly specifying them in And-Or Graphs. This limits the scalability of these explanations in terms of porting them to a new domain/task. In this Chapter, we propose a conceptual and counterfactual explanation framework for explaining decisions made by a deep convolutional neural network (CNN) [AWZ20]. Unlike X-ToM, we do not assume any underlying representations for the explanation parse graph in this Chapter. Instead, we learn the high-level concepts semi-automatically from the training dataset.

3.1 Introduction

we present a new XAI model CoCoX which explains decisions made by a deep convolutional neural network (CNN) using *fault-lines* [KT81].

Fault-lines are the high-level semantic aspects of reality that humans zoom in on when they imagine an alternative to it. More concretely, given an input image I for which a CNN model M predicts class c_{pred}, our fault-line based explanation identifies a *minimal* set of semantic features, referred to as *explainable concepts* (xconcepts), that need to be added to or deleted from I in order to alter the classification category of I by M to another specified class c_{alt}. For example, let us consider a training dataset for an image classification task shown in Figure 1.3 containing the classes

Dog, Thylacine, Frog, Toad, Goat and Sheep, and a CNN based classification model M which is trained on this dataset. In order to alter the model's prediction of input image I_1 from Dog to Thylacine, the fault-line ($\Psi^{+}_{I_1,c_{pred},c_{alt}}$) suggests adding *stripes* to the Dog. We call this a positive fault-line (PFT) as it involves adding a new xconcept, i.e., *stripedness*, to the input image. Similarly, to change the model prediction of I_2 from Toad to Frog, the fault-line ($\Psi^{-}_{I_2,c_{pred},c_{alt}}$) suggests removing *bumps* from the Toad. We call this a negative fault-line (NFT) as it involves subtracting xconcept, i.e., *bumpedness*, from the input image. In most cases, both PFT and NFT are needed to successfully alter the model prediction.

For example, in Figure 1.3, in order to change the model prediction of I_3 from Goat to Sheep, we need to add an xconcept *wool* (PFT) to I_3 and also remove xconcepts *beard* and *horns* (NFT) from I_3. As we can see, these fault-lines can be directly used to make the internal decision making criteria of deep neural network transparent to both expert and non-expert users. For instance, we answer the question *"Why does the machine classify the image I_3 as Goat instead of Sheep?"* by using PFT $\Psi^{+}_{I_3,c_{pred},c_{alt}}$ and NFT $\Psi^{-}_{I_3,c_{pred},c_{alt}}$ as follows: "Machine thinks the input image is Goat and not Sheep mainly because Sheep's feature *woolly* is absent in I_3 and Goat's features beard and horns are present in I_3". It may be noted that there could be several other features of Sheep and Goat that might have influenced the model's prediction. However, fault-lines only capture the most critical (minimal) features that highly influenced the model's prediction.

What makes fault-lines a good visual explanation? We chose fault-lines as an explanation for the following two important reasons:

1. Firstly, unlike current methods in XAI which mainly focus on pixel-level explanations (viz. saliency maps), fault-line based explanations are **concept-level** explanations. Pixel-level explanations are not effective at human scale, whereas concept level explanations are effective, less ambiguous, and more natural for both expert and non-expert users in building a mental model of a vision system [KWG18]. Moreover, with conceptual explanations, humans can easily generalize their understanding to new unseen instances/tasks. In our work, as shown in Figure 1.3, we represent xconcepts (e.g., *stripedness*) using a set of example images (similar to [KWG18]).

2. Secondly, fault-lines are **counter-factual** in nature, i.e., they provide a *minimal* amount of information capable of altering a decision. This makes them easily digestible and practically useful for understanding the reasons for a model's decision [WMR17]. For example, consider the fault-line explanation for image I_3 in Figure 1.3. The explanation provides only the most critical changes (i.e., adding wool and removing beard and horns) required to alter the model's prediction from `Goat` to `Sheep`, though several other changes may be necessary.

While there are recent works on generating pixel-level counter-factual and contrastive explanations [HHD18, DCL18, GWE19], to the best of our knowledge, this is the first work to propose a method for generating explanations that are counter-factual as well as conceptual.

We identify two main challenges in generating a fault-line explanation, namely: (a) How to identify the set of xconcepts; and (b) How to select the most critical xconcepts that alter the model prediction from c_{pred} to c_{alt}. In this work, we first propose a novel method to mine all the plausible xconcepts from the given dataset automatically. We then identify class-specific xconcepts by using directional derivatives [KWG18]. Finally, we pose the derivation of a fault-line as an optimization problem which selects a minimal set of these xconcepts to alter the model's prediction. We perform extensive human study experiments to demonstrate the effectiveness of our approach in improving human understanding of the underlying classification model.

Through our human studies, we show that our fault-line based explanations significantly outperform the baselines (i.e., attribution techniques and pixel-level counterfactual explanations) in terms of qualitative and quantitative metrics such as Justified Trust and Explanation Satisfaction [HMK18].

Concurrent to our work, recent work by [GWK19] also seeks to automatically identify human-friendly xconcepts. However, they use segmentation methods to identify xconcepts, whereas we use Grad-CAM [SCD17a] based localization maps. Moreover, their explanations are not counter-factual unlike our fault-line based explanations.

The contributions of this Chapter are threefold: (i) we introduce a new XAI framework based on fault-lines to generate conceptual and counterfactual explanations; (ii) we present a new method to

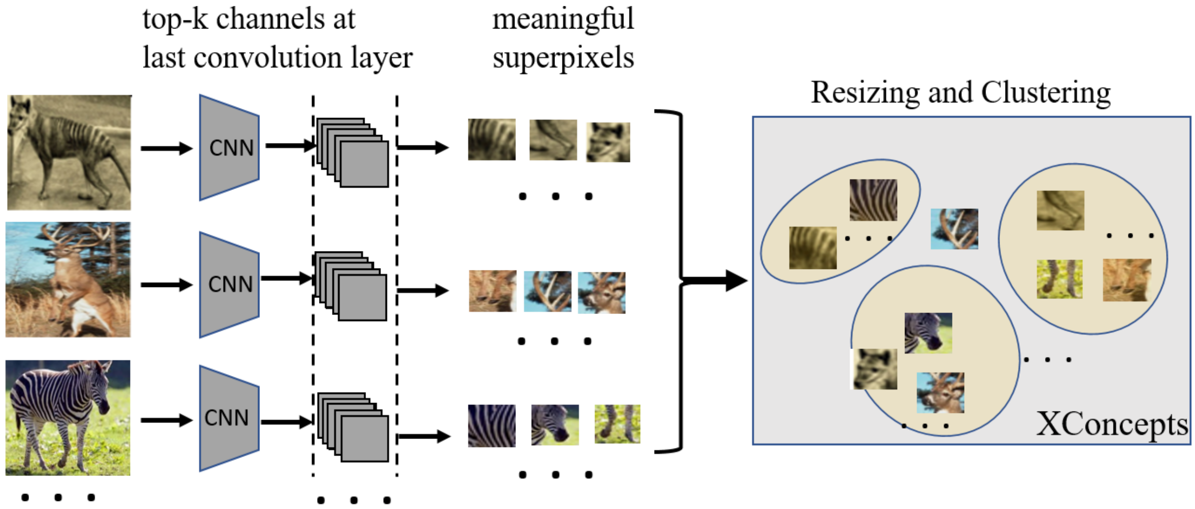


Figure 3.1: We consider feature maps from the last convolutional layer as instances of xconcepts and obtain their localization maps (i.e., superpixels) by computing the gradients of the output with respect to the feature maps. We select highly influential superpixels and then apply K-means clustering with outlier removal to group these superpixels into clusters where each cluster represents an xconcept.

mine xconcepts from a given training dataset automatically and derive the fault-lines; (iii) we show that our fault-line explanations qualitatively and quantitatively outperform baselines in improving human understanding of the classification model.

3.2 Approach

In this section, we detail our ideas and methods for generating fault-line explanations. Without loss of generality, we consider a pre-trained CNN (M) for image classification. Given an input image I, the CNN predicts a log-probability output $\log P\left(Y|I\right)$ over the output classes Y. Let χ denote a dataset of training images, where $\chi_c \subset \chi$ represents the subset that belongs to category $c \in Y$, $(c = 1, 2, \ldots, C)$. We denote the score (logit) for class c (before the softmax) as y^c and the predicted class label as c_{pred}. Our high-level goal is to find a fault-line explanation (Ψ) that alters the CNN prediction from c_{pred} to another specified class c_{alt} using a minimal number of xconcepts. We follow [KWG18] in defining the notion of xconcepts where each xconcept is represented using a set of example images. This representation of xconcepts provides great flexibility and portability as it will not be constrained to input features or a training dataset, and one can utilize the generated xconcepts across multiple datasets and tasks.

We represent the quadruple $<I,\ c_{pred},\ c_{alt}>$ as a human's query Q that will be answered by showing a fault-line explanation Ψ. We use Σ to represent all the xconcepts mined from χ. The xconcepts specific to the class c_{pred} and c_{alt} are represented as Σ_{pred} and Σ_{alt} respectively. Our strategy will be to first identify the xconcepts Σ_{pred} and Σ_{alt} and then generate a fault-line explanation by finding a minimal set of xconcepts from Σ_{pred} and Σ_{alt}. Formally, the objective is to find a fault-line that maximizes the posterior probability:

$$\arg\max_{\Psi} P\left(\Psi, \Sigma_{pred}, \Sigma_{alt}, \Sigma \,\middle|\, Q\right) \tag{3.1}$$

3.2.1 Mining Xconcepts

We first compute $P\left(\Sigma \,|\, \chi, M\right)$ by identifying a set of semantically meaningful superpixels from every image and then perform clustering such that all the superpixels in a cluster are semantically similar. Each of these clusters represent an xconcept. We then identify class specific xconcepts i.e., $P\left(\Sigma_{pred} \,\middle|\, \Sigma, \chi, I, c_{pred}, M\right)$ and $P\left(\Sigma_{alt} \,|\, \Sigma, \chi, I, c_{alt}, M\right)$.

3.2.1.1 A. Finding Semantically Meaningful Super-pixels as Xconcepts

Figure 5.10 shows the overall algorithm for computing $P\left(\Sigma \,|\, \chi, M\right)$. As deeper layers of the CNN capture richer semantic aspects of the image, we construct the xconcepts by making use of feature maps from the last convolution layer. Let f denote the feature extractor component of the CNN and g denote the classifier component of the CNN that takes the output of f and predicts log-probabilities over output classes Y. We denote the m feature maps produced at layer L of the CNN as $A^{m,L} = \{a^L | a^L = f(I)\}$ which are of width u and height v. We consider each feature map as an instance of an xconcept and obtain its localization map (i.e., super-pixels of each feature map). To produce the localization map, we use Grad-CAM [SCD17a] to compute the gradients of y^c with respect to the feature maps $A^{m,L}$ and are then spatially pooled using Global Average Pooling (GAP) to obtain the importance weights ($\alpha^c_{m,L}$) of a feature map m at layer L for a target class c:

$$\alpha^c_{m,L} = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^{m,L}_{ij}} \tag{3.2}$$

Using the importance weights, we select top p super-pixels for each class. Given that there are C output classes in the dataset $\mathcal{X}$, we get $p * C$ super-pixels from each image in the training dataset. We apply K-means clustering with outlier removal to group these super-pixels into G clusters where each cluster represents an xconcept (as shown in Figure 5.10). For clustering, we consider the spatial feature maps $f(I)$ instead of the super-pixels (i.e., actual image regions) themselves. We use the silhouette score value of a different range of clusters to determine the value of K.

3.2.1.2 B. Identifying Class-Specific Xconcepts

For each output class c, we learn the most common xconcepts that are highly influential in the prediction of that class over the entire training dataset $\mathcal{X}$. We use the TCAV technique [KWG18] to identify these class-specific xconcepts. Specifically, we construct a vector representation of each xconcept, called a CAV (denoted as v_X), by using a direction normal to a linear classifier trained to distinguish between the xconcept activations from the random activations. We then compute directional derivatives ($S_{c,X}$) to produce estimates of how important the concept X was for a CNN's prediction of a target class c, e.g., how important the xconcept `stripedness` is for predicting the zebra class.

$$S_{c,X} = \nabla g_c(f(I)) \cdot v_X \tag{3.3}$$

where g_c denote the classifier component of the CNN that takes the output of f and predicts log-probability of output class c. We argue that these class-specific xconcepts facilitate in generating meaningful explanations by pruning out incoherent xconcepts. For example, the xconcepts such as `wheel` and `wings` are irrelevant in explaining why the network's prediction is a $zebra$ and not a

$cat.$

3.2.2 Fault-Line Identification

In this subsection, we describe our approach to generate a fault-line explanation using the class-specific xconcepts.

Let us consider that n_{pred} and n_{alt} xconcepts have been identified for output classes c_{pred} and c_{alt} respectively, i.e., $\left|\Sigma_{pred}\right| = n_{pred}$ and $|\Sigma_{alt}| = n_{alt}$. We denote CAVs of the n_{pred} xconcepts belonging to the class c_{pred} as $v_{pred} = \{v^i_{pred},\ i = 1, 2, \ldots, n_{pred}\}$ and CAVs of the n_{alt} xconcepts belonging to the class c_{alt} as $v_{alt} = \{v^i_{alt},\ i = 1, 2, \ldots, n_{alt}\}$. We formulate finding a fault-line explanation as the following optimization problem:

$$\begin{aligned}
&\underset{\delta_{pred},\delta_{alt}}{\text{minimize}} \quad \alpha D(\delta_{pred}, \delta_{alt}) + \beta \left\|\delta_{pred}\right\|_1 + \lambda \left\|\delta_{alt}\right\|_1 ; \\
&D(\delta_{pred}, \delta_{alt}) = max\{g^{pred}(I^{'}) - g^{alt}(I^{'}), -\tau\}; \\
&I^{'} = A^{m,L} \circ v^{\top}_{pred}\delta_{pred} \circ v^{\top}_{alt}\delta_{alt}; \\
&\delta^i_{pred} \in \{-1, 0\},\ \delta^i_{alt} \in \{0, 1\}\ \forall i \text{ and } \alpha, \beta, \lambda, \tau \geq 0.
\end{aligned} \tag{3.4}$$

We elaborate on the role of each term in the Equation 3.4 as follows. Our goal here is to derive a fault-line explanation that gives us the minimal set of xconcepts from Σ_{pred} and Σ_{alt} that will alter the model prediction from c_{pred} to c_{alt}. Intuitively, we try creating new images (I') by removing xconcepts in Σ_{pred} from I and adding xconcepts in Σ_{alt} to I until the classification result changes from c_{pred} to c_{alt}. To do this, we do not directly perturb the original image but change the activations obtained at last convolutional layer $A^{m,L}$ instead. In order to perturb the activations, we take the Hadamard product ($\circ$) between the activations ($A^{m,L}$), $v^{\top}_{pred}\delta_{pred}$ and $v^{\top}_{alt}\delta_{alt}$. The difference between the new logit scores for c_{pred} (i.e., $g^{pred}(I')$) and c_{alt} (i.e,. $g^{alt}(I')$) is controlled by the parameter τ. We apply a projected fast iterative shrinkage-thresholding algorithm (FISTA) [BT09, DCL18] for solving the above optimization problem. We outline our method in Algorithm 1.

Algorithm 1: Generating Fault-Line Explanations

```
input image $I$, classification model $M$, predicted class label $c_{pred}$, alternate class label $c_{alt}$ and training dataset $\chi$
1. Find semantically meaningful superpixels in $\chi$,
       $\alpha^{c}_{m,L} = \frac{1}{Z} \sum_{i} \sum_{j} \frac{\partial y^{c}}{\partial A^{m,L}_{ij}}$
2. Apply K-means clustering on superpixels and obtain xconcepts ($\Sigma$).
3. Identify class specific xconcepts ($\Sigma_{pred}$ and $\Sigma_{alt}$) using TCAV,
       $S_{c,X} = \nabla g_{c}(f(I)) \cdot v_{X}$
4. Solve Equation 3.4 to obtain fault-line $\Psi$,
       $\Psi \leftarrow \min_{\delta_{pred},\delta_{alt}} \alpha D(\delta_{pred}, \delta_{alt}) + \beta \left\|\delta_{pred}\right\|_{1} + \lambda \left\|\delta_{alt}\right\|_{1}$
return $\Psi$.
```

3.3 Experiments

We conducted extensive human subject experiments to quantitatively and qualitatively assess the effectiveness of the proposed fault-line explanations in helping expert human users and non-expert human users understand the internal workings of the underlying model. We chose an image classification task for our experiments (although the proposed approach is generic and can be applied to any task). We use the following metrics [HMK18, Hof17b] to compare our method with the baselines[*].

1. ***Justified Trust*** (Quantitative Metric). Justified Trust is computed by evaluating the human's understanding of the model's (M) decision-making process. In other words, given an image, it evaluates whether the users could reliably predict the model's output decision. More concretely, let us consider that M predicts images in a set C correctly and makes incorrect decisions on the images in the set W. Justified trust is given as sum of the percentage of images in C that the human subject thinks M would correctly predict and the percentage of images in W that the human subject thinks M would fail to predict correctly.

2. ***Explanation Satisfaction (ES)*** (Qualitative Metric). We measure human subjects' feeling of satisfaction at having achieved an understanding of the machine in terms of usefulness, sufficiency, appropriated detail, confidence, and accuracy [HMK18, Hof17b]. We ask the subjects to rate each of these metrics on a Likert scale of 0 to 9.

We used ILSVRC2012 dataset (Imagenet) [RDS15] and considered VGG-16 [SZ14] as the underlying network model. We randomly chose 40 classes in the dataset for our experiments and identified 46 xconcepts using our algorithm[†].

We applied between-subject design and randomly assigned subjects into ten groups. We perform

[*] We empirically observed that the metrics Justified Trust and Explanation Satisfaction are effective in evaluating the core objective of XAI, i.e. to evaluate whether the user's understanding of the model improves with explanations. These metrics are originally defined at a high-level in the work by [HMK18] and we adapt them for the image classification

XAI Framework	**Justified Trust** (±std)	**Explanation Satisfaction** (±std)				
		Confidence	Usefulness	Appropriate Detail	Understandability	Sufficiency
NO-X	21.4% ± 2.7%	N/A	N/A	N/A	N/A	N/A
CAM [ZKL16]	24.0% ± 1.9%	4.2 ± 1.8	3.6 ± 0.8	2.2 ± 1.9	3.2 ± 0.9	2.6 ± 1.3
Grad-CAM [SCD17a]	29.2% ± 3.1%	4.1 ± 1.1	3.2 ± 1.9	3.0 ± 1.6	4.2 ± 1.1	3.2 ± 1.0
LIME [RSG16]	46.1% ± 1.2%	5.1 ± 1.8	4.2 ± 1.6	3.9 ± 1.1	4.1 ± 2.0	4.3 ± 1.6
LRP [BBM15]	31.1% ± 2.5%	1.1 ± 2.2	2.8 ± 1.0	1.6 ± 1.7	2.8 ± 1.0	2.1 ± 1.8
SmoothGrad [STK17]	37.6% ± 2.9%	1.4 ± 1.0	2.2 ± 1.8	2.8 ± 1.0	3.1 ± 0.8	2.9 ± 0.8
TCAV [KWG18]	49.7% ± 3.3%	3.6 ± 2.1	3.2 ± 1.8	3.3 ± 1.6	3.6 ± 2.1	3.9 ± 1.1
CEM [DCL18]	51.0% ± 2.1%	4.1 ± 1.4	3.4 ± 1.4	3.1 ± 2.1	2.9 ± 0.9	3.3 ± 1.6
CVE [GWE19]	50.9% ± 3.0%	3.8 ± 1.9	3.1 ± 0.9	3.6 ± 2.1	4.1 ± 1.2	4.2 ± 1.2
CoCoX (Fault-lines)	**69.1% ± 2.1%**	**6.2 ± 1.2**	**6.6 ± 0.7**	**7.2 ± 0.9**	**7.1 ± 0.6**	**6.2 ± 0.8**

Table 3.1: Quantitative (Justified Trust) and Qualitative (Explanation Satisfaction) comparison of CoCoX on Non-Expert Pool with random guessing baseline, no explanation (NO-X) baseline, and other state-of-the-art XAI frameworks such as CAM, Grad-CAM, LIME, LRP, SmoothGrad, TCAV, CEM, and CVE.

this separately with expert user pool and non-expert user pool. Subjects in non-expert pool have no background in computer vision, whereas subjects in expert pool are experienced in training an image classification model using CNN. Each group in the non-expert pool are assigned 6 subjects and each group in the expert pool are assigned 2 subjects. Within each group, each subject will

task.

[†] We manually removed noisy xconcepts and fault-lines. We couldn't find an automatic approach to filter them. We found that xconcepts generated by [GWK19] are less noisy and might help in generating more meaningful fault-lines. We leave this for future exploration.

XAI Framework	**Justified Trust** (±std)	**Explanation Satisfaction** (±std)				
		Confidence	Usefulness	Appropriate Detail	Understandability	Sufficiency
NO-X	$28.1\,\% \pm 4.1\,\%$	N/A	N/A	N/A	N/A	N/A
CAM [ZKL16]	$37.1\,\% \pm 3.9\,\%$	3.2 ± 1.8	3.3 ± 1.4	3.1 ± 2.1	3.1 ± 1.8	2.9 ± 1.9
Grad-CAM [SCD17a]	$39.1\,\% \pm 2.1\,\%$	3.7 ± 1.2	3.1 ± 2.2	2.7 ± 1.9	3.7 ± 1.1	3.4 ± 1.6
LIME [RSG16]	$42.1\,\% \pm 3.1\,\%$	3.1 ± 2.2	3.0 ± 1.2	2.8 ± 1.9	3.1 ± 2.2	2.8 ± 1.7
LRP [BBM15]	$51.1\,\% \pm 3.1\,\%$	3.2 ± 4.1	3.5 ± 1.6	4.2 ± 1.5	4.3 ± 1.0	3.9 ± 0.9
SmoothGrad [STK17]	$40.7\,\% \pm 2.1\,\%$	3.1 ± 1.0	2.9 ± 1.2	3.8 ± 1.5	3.3 ± 1.1	3.1 ± 1.0
TCAV [KWG18]	$55.1\,\% \pm 3.3\,\%$	3.9 ± 2.8	3.6 ± 1.6	4.1 ± 1.3	4.9 ± 1.2	3.9 ± 0.8
CEM [DCL18]	$61.1\,\% \pm 2.2\,\%$	4.8 ± 1.6	3.7 ± 1.6	4.0 ± 1.2	3.7 ± 1.0	4.0 ± 1.1
CVE [GWE19]	$64.5\,\% \pm 3.7\,\%$	4.1 ± 2.3	3.9 ± 1.5	4.6 ± 1.5	4.5 ± 1.4	3.9 ± 1.2
CoCoX (Fault-lines)	$\mathbf{70.5\,\% \pm 1.3\,\%}$	$\mathbf{5.7 \pm 1.1}$	$\mathbf{4.9 \pm 0.8}$	$\mathbf{5.8 \pm 1.2}$	$\mathbf{6.9 \pm 1.1}$	$\mathbf{6.4 \pm 1.0}$

Table 3.2: Performance comparison on expert subject pool.

first go through a familiarization phase where the subjects become familiar with the underlying model through explanations (with 15 training images), followed by a testing phase where we apply our evaluation metrics and assess their understanding (on 5 test images) in the underlying model. Specifically, in the familiarization phase, human will be shown the input image I and the CNN's prediction c_{pred} and asked to provide c_{alt} as input. We will then show an explanation to the human user for the model's prediction c_{pred}. For example, in CoCoX group, we show the fault-line explaining why the model chose c_{pred} instead of c_{alt}. In the testing phase, human will be given only I and will not see c_{pred}, c_{alt}, and explanations, and we evaluate whether the human can correctly identify c_{pred} based on his/her understanding of the model gained in the familiarization phase.

For the first group, called NO-X (short for no-explanation group), we show the model's classifi-

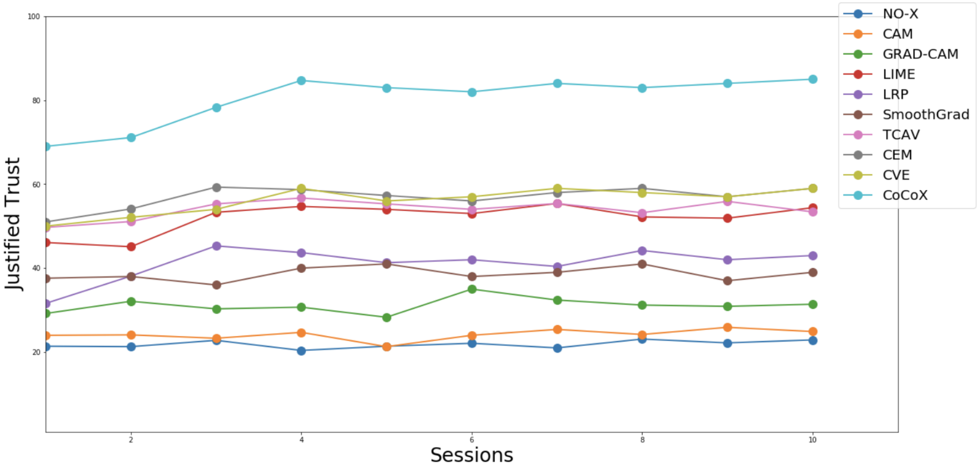


Figure 3.2: Gain in Justified Trust over time.

cation output on all the 15 images in the familiarization phase but we do not provide any explanation for the model's prediction. For the subjects in groups two to nine, in addition to the model's classification output, we also provide explanations in the familiarization phase for the model's prediction generated using the following state-of-the-art XAI models respectively: CAM [ZKL16], Grad-CAM [SCD17a], LIME [RSG16], LRP [BBM15], SmoothGrad [STK17], TCAV [KWG18], CEM [DCL18], and CVE [GWE19]. For the subjects in the tenth group, we show the fault-line explanations generated by our CoCoX model in addition to the classification output. It may be noted that, in the testing phase, human will be shown only the image I and will not be provided c_{pred}, c_{alt}, and explanations.

3.3.1 Results

Table 3.1 and Table 3.2 compares the Justified Trust (JT) and Explanation Satisfaction (ES) of all the ten groups in expert subject pool and non-expert subject pool. As we can see, JT and ES values of attention map based explanations such as Grad-CAM, CAM, and SmoothGrad do not differ significantly from the NO-X baseline, i.e., attention based explanations are not effective at increasing human trust and reliance (we did not evaluate ES for NO-X group as these subjects are not

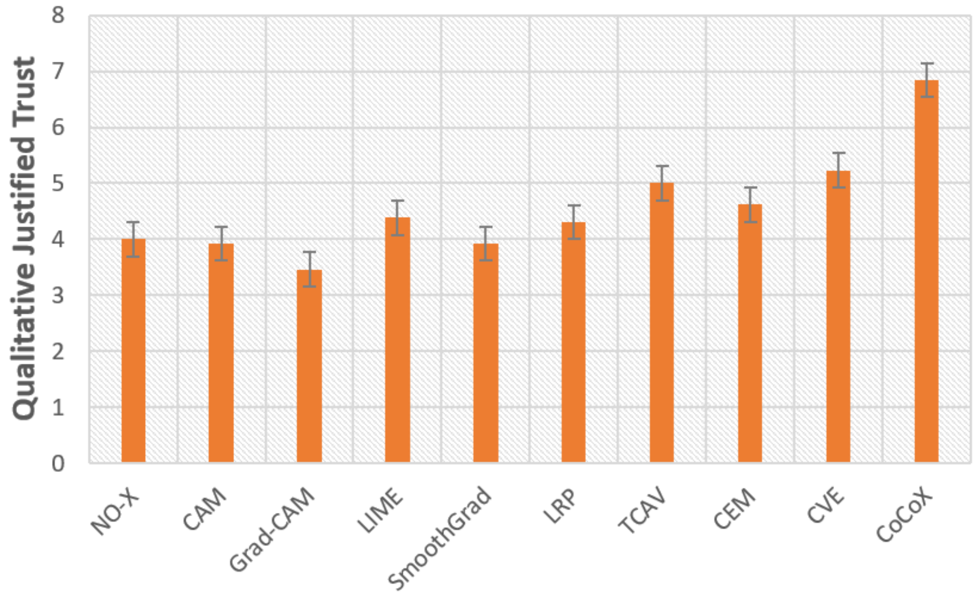


Figure 3.3: Average Qualitative Justified Trust (on a Likert scale of 0 to 9). Error bars denote standard errors of the means.

shown any explanations). This finding is consistent with the recent study by [JW19] which shows that attention is not an explanation. On the other hand, concept based explanation framework TCAV and counterfactual explanation frameworks CEM, and CVE performed significantly better than the NO-X baseline (in both expert and non-expert pool). Our CoCoX model, which is both conceptual and counterfactual, significantly outperformed all the baselines with 69.1% JT in non-expert pool and 70.5% JT in expert pool ($p < 0.01$). Interestingly, expert users preferred LRP (JT = 51.1%) to LIME (JT = 42.1%) and non-expert users preferred LIME (JT = 46.1%) to LRP (JT = 31.1%).

Furthermore, human subjects in the CoCoX group, compared to all the other baselines, found that explanations are highly useful, sufficient, understandable, detailed and are more confident in answering the questions in the testing phase. These findings verify our hypothesis that fault-line explanations are lucid and easy for both expert and non-expert users to understand.

Gain in Justified Trust over Time: We hypothesized that subjects' justified trust in the AI system might improve over time. This is because it can be harder for humans to fully understand the

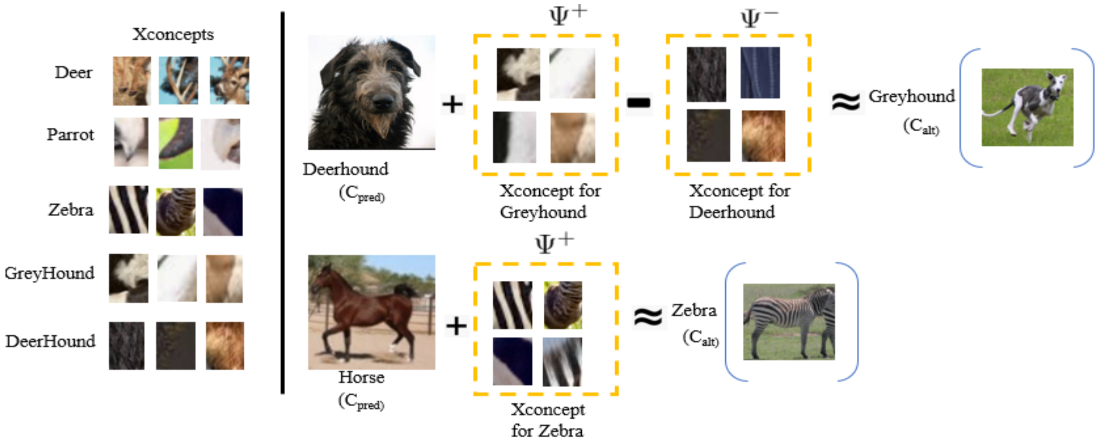


Figure 3.4: Examples of xconcepts (**Left**) and fault-line explanations (**Right**) identified by our method.

machine's underlying inference process in one single session. Therefore, we conduct an additional experiment with eight human subjects (non-experts) for each group where the subjects' reliance was measured after every session. Note that each session consists of a familiarization phase followed by a testing phase. The results are shown in Figure 3.2. As we can see, the subjects' JT in CoCoX group increased at a higher-rate compared to other baselines. However, we did not find any significant increase in JT after fifth session across all the groups. This is consistent with our expectation that it is difficult for humans to focus on a task for longer periods [‡]. It should be noted that the increase in JT with attention map based explanations such as Grad-CAM and CAM is not significant. This finding again demonstrates that attention maps are not effective to improve human trust.

Subjective Evaluation of Justified Trust: In addition to the quantitative evaluation of the justified trust, we also collect subjective trust values (on a Likert scale of 0 to 9) from the subjects. This helps in understanding to what extent the users think they trust the AI system. The results are shown in Figure 3.3. As we can see, these results are consistent with our quantitative trust measures except that qualitative trust in Grad-CAM, CAM, and SmoothGrad is lower compared to the NO-X

[‡] In the future, we also intend to experiment with subjects by arranging sessions over days or weeks instead of having continuous back to back sessions.

group.

Case Study: Figure 3.4 shows examples of the xconcepts (cropped and rescaled for better view) identified using our approach. As we can see, our method successfully extracts semantically coherent xconcepts such as *pointed curves* of `deer`, *stripedness* of `zebra`, and *woolliness* of `deerhound` from the training dataset. Also the fault-lines generated by our method correctly identify the most critical xconcepts that can alter the classification result from c_{pred} to c_{alt}. For example, consider the image of `deerhound` shown in the Figure 3.4. Our fault-line explanation suggests removing *woolliness* and adding *black and white pattern* to alter the model's classification on the image from `deerhound` to `greyhound`.

3.4 Related Work

Most prior work has focused on generating explanations using feature visualization and attribution. **Feature visualization** techniques typically identify qualitative interpretations of features used for making predictions or decisions. For example, gradient ascent optimization is used in the image space to visualize the hidden feature layers of unsupervised deep architectures [EBC09]. Also, convolutional layers are visualized by reconstructing the input of each layer from its output [ZF14]. Recent visual explanation models seek to jointly classify the image and explain why the predicted class label is appropriate for the image [HAR16]. Other related work includes a visualization-based explanation framework for Naive Bayes classifiers [SGL03], an interpretable character-level language models for analyzing the predictions in RNNs [KJF15], and an interactive visualization for facilitating analysis of RNN hidden states [SGH16].

Attribution is a set of techniques that highlight pixels of the input image (saliency maps) that most caused the output classification. Gradient-based visualization methods [ZKL16, SCD17a] have been proposed to extract image regions responsible for the network output. The LIME method proposed by [RSG16] explains predictions of any classifier by approximating it locally with an interpretable model.

There are few recent works in the XAI literature that go beyond the pixel-level explanations. For example, the TCAV technique proposed by [KWG18] aims to generate explanations based on high-level user defined concepts. Contrastive explanations are proposed by [DCL18] to identify minimal and sufficient features to justify the classification result. [GWE19] proposed counterfactual visual explanations that identify how the input could change such that the underlying vision system would make a different decision. More recently, few methods have been developed for building models which are intrinsically interpretable [ZNZ18]. In addition, there are several works [Mil18b] on the goodness measures of explanations which aim to assess the underlying characteristics of explanations.

3.5 Summary

In this Chapter, we introduced a new explainable AI (XAI) framework, CoCoX, based on fault-lines. We argue that due to their conceptual and counterfactual nature, fault-line based explanations are lucid, clear and easy for humans to understand. We proposed a new method to automatically mine explainable concepts from a given training dataset and to derive fault-line explanations. Using qualitative and quantitative evaluation metrics, we demonstrated that fault-lines significantly outperform baselines in improving human understanding of the underlying classification model.

3.6 Appendix

3.6.1 More Examples of our Extracted Xconcepts

We provide more examples of the extracted xconcepts (along with the original image for clarity) in Figure 3.5.

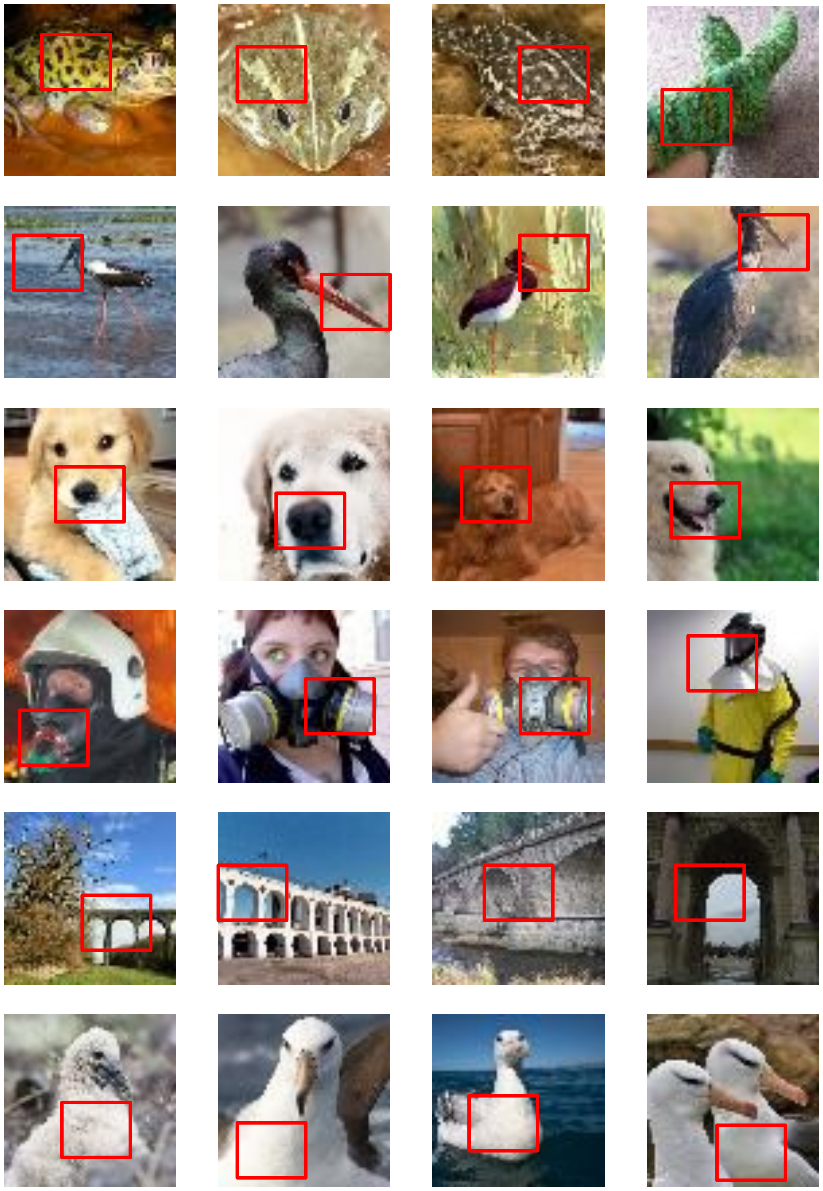


Figure 3.5: More examples for the Xconcepts.

CHAPTER 4

Explaining Model Biases and Improving Robustness

In Chapter 1 and 2, we proposed explanation frameworks to help users understand the model decisions. In this Chapter, we critically examine state-of-the-art models and benchmarks for vision and langauge grounding tasks to evaluate the extent to which these models are interpretable, faithful and robust to out-of-distribution and adversarial samples [AGA20a]. In addition, we also propose new methods to improve robustness and compositional reasoning skills of these models.

4.1 Introduction

In this Chapter, we consider the task of visual referring expression recognition to conduct our evaluation. Visual referring expression recognition is the task of identifying the object in an image referred by a natural language expression [KOM14, NMD16, MHT16, HXR16]. Figure 4.1 shows an example. This task has drawn much attention due to its ability to test a model's understanding of natural language in the context of visual grounding and its application in downstream tasks such as image retrieval [YLH14] and question answering [AAL15, ZGB16, GAM12, ASM13, PRA15, Aku15, AZ19]. To track progress on this task, various datasets have been proposed, in which real world images are annotated by crowdsourced workers [KOM14, MHT16]. Recently, neural models have achieved tremendous progress on these datasets [YLS18, LBP19a]. However, multiple studies have suggested that these models could be exploiting strong biases in these datasets [CMB18, LLB19]. For example, models could be just selecting a salient object in an image or a referring expression without recourse to linguistic structure (see Figure 4.1). This defeats the true purpose of the task casting doubts on the actual progress.

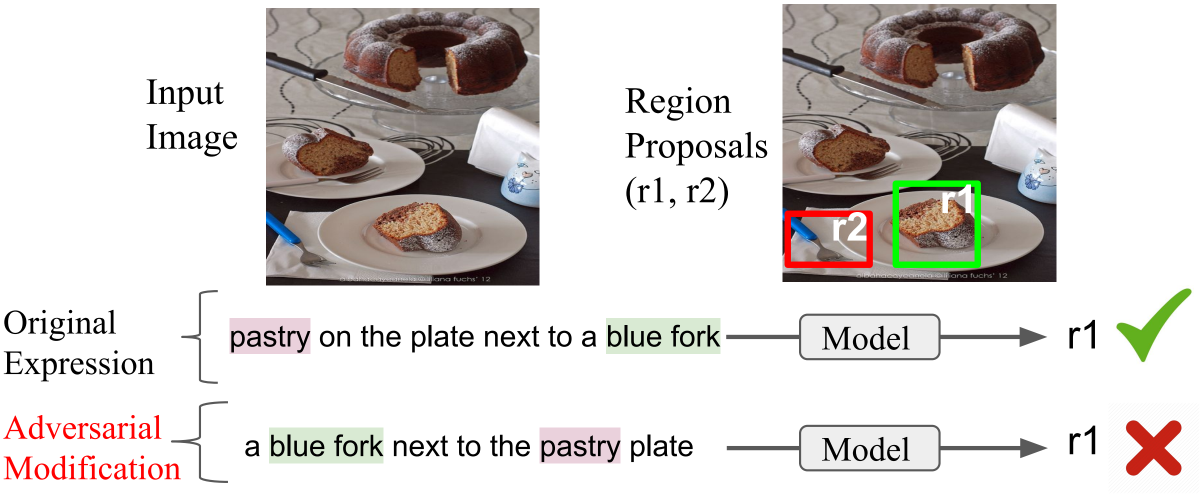

Input Image
Region Proposals (r1, r2)
r2
r1
Original Expression
pastry on the plate next to a blue fork
Model
r1
Adversarial Modification
a blue fork next to the pastry plate
Model
r1

Figure 4.1: An example of the visual referring expression recognition task. If the word *pastry* is present in the referring expression, models prefer the bounding box *r1* (highlighted in green) irrespective of the change in linguistic structure (word order).

In this Chapter, we examine *RefCOCOg* dataset [MHT16], a popular testbed for evaluating referring expression models, using crowdsourced workers. We show that a large percentage of samples in the *RefCOCOg* test set indeed do not rely on linguistic structure (word order) of the expressions. Accordingly, we split *RefCOCOg* test set into two splits, *Ref-Easy* and *Ref-Hard* , where linguistic structure is key for recognition in the latter but not the former (§4.2). In addition, we create a new out-of-distribution* dataset called *Ref-Adv* using *Ref-Hard* by rewriting a referring expression such that the target object is different from the original annotation (§4.3). We evaluate existing models on these splits and show that the true progress is at least 12-23% behind the established progress, indicating there is ample room for improvement (§4.4). We propose two new models, one which make use of contrastive learning using negative examples, and the other based on multi-task learning, and show that these are slightly more robust than the current state-of-the-art models (§4.5).

*This is a *contrast set* according to [GAB20]

4.2 Importance of linguistic structure

RefCOCOg is the largest visual referring expression benchmark available for real world images [MHT16]. Unlike other referring expression datasets such as *RefCOCO* and *RefCOCO+* [KOM14], a special care has been taken such that expressions are longer and diverse. We therefore choose to examine the importance of linguistic structure in *RefCOCOg* . CirikMB18 observed that when the words in a referring expression are shuffled in random order, the performance of existing models on *RefCOCOg* drops only a little. This suggests that models are relying heavily on the biases in the data than on linguistic structure, i.e., the actual sequence of words. Ideally, we want to test models on samples where there is correlation between linguistic structure and spatial relations of objects, and any obscurity in the structure should lead to ambiguity. To filter out such set, we use humans.

We randomly shuffle words in a referring expression to distort its linguistic structure, and ask humans to identify the target object of interest via predefined bounding boxes. Each image in *RefCOCOg* test set is annotated by five Amazon Mechanical Turk (AMT) workers and when at least three annotators select a bounding box that has high overlap with the ground truth, we treat it as a correct prediction. Following [MHT16], we set 0.5 IoU (intersection over union) as the threshold for high overlap.

Given that there are at least two objects in each image, the optimal performance of a random choice is less than 50%.[†] However, we observe that human accuracy on distorted examples is 83.7%, indicating that a large portion of *RefCOCOg* test set is insensitive to linguistic structure. Based on this observation, we divide the test set into two splits for fine-grained evaluation of models: ***Ref-Easy*** contains samples insensitive to linguistic structure and ***Ref-Hard*** contains sensitive samples (statistics of the splits are shown in Table 6.6).

[†] On average, there are 8.2 bounding boxes per image.

	Ref-Easy	*Ref-Hard*	*Ref-Adv*
data size	8034 (83.7% of *RefCOCOg*)	1568 (16.3% of *RefCOCOg*)	3704
avg. length in words	8.0	10.2	11.4

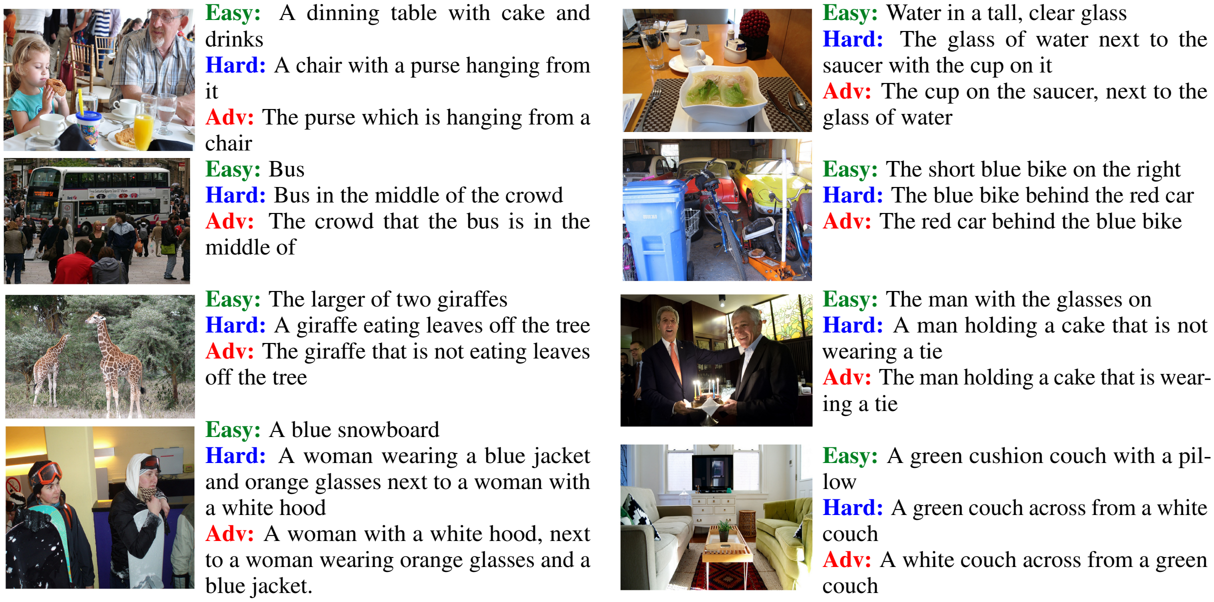
Table 4.1: Statistics of *Ref-Easy* , *Ref-Hard* and *Ref-Adv* . *Ref-Easy* and *Ref-Hard* indicate the proportion of samples in *RefCOCOg* test set that are insensitive and sensitive to linguistic structure respectively.

4.3 An out-of-distribution dataset

Due to unintended annotation artifacts in *RefCOCOg* , it is still possible that models could perform well on *Ref-Hard* without having to rely on linguistic structure, e.g., by selecting frequent objects seen during training time. Essentially, *Ref-Hard* is an in-distribution split. To avoid this, we create ***Ref-Adv*** , an adversarial test set with samples that may be fall out of training distribution.

We take each sample in *Ref-Hard* and collect additional referring expressions such that the target object is different from the original object. We chose the target objects which humans are most confused with when the referring expression is shuffled (as described in the previous section). For each target object, we ask three AMT workers to write a referring expression while retaining most content words in the original referring expression. In contrast to the original expression, the modified expression mainly differs in terms of the structure while sharing several words. For example, in Figure 4.1, the adversarial sample is created by swapping *pastry* and *blue fork* and making *plate* as the head of *pastry*. We perform an extra validation step to filter out bad referring expressions. In this step, three additional AMT workers select a bounding box to identify the target object, and we only select the samples where at least two workers achieve IoU > 0.5 with the target object.

Since the samples in *Ref-Adv* mainly differ in linguistic structure with respect to *Ref-Hard* , we hope that a model which does not make use of linguistic structure (and correspondingly spatial



Easy: A dinning table with cake and drinks
Hard: A chair with a purse hanging from it
Adv: The purse which is hanging from a chair

Easy: Bus
Hard: Bus in the middle of the crowd
Adv: The crowd that the bus is in the middle of

Easy: The larger of two giraffes
Hard: A giraffe eating leaves off the tree
Adv: The giraffe that is not eating leaves off the tree

Easy: A blue snowboard
Hard: A woman wearing a blue jacket and orange glasses next to a woman with a white hood
Adv: A woman with a white hood, next to a woman wearing orange glasses and a blue jacket.

Easy: Water in a tall, clear glass
Hard: The glass of water next to the saucer with the cup on it
Adv: The cup on the saucer, next to the glass of water

Easy: The short blue bike on the right
Hard: The blue bike behind the red car
Adv: The red car behind the blue bike

Easy: The man with the glasses on
Hard: A man holding a cake that is not wearing a tie
Adv: The man holding a cake that is wearing a tie

Easy: A green cushion couch with a pillow
Hard: A green couch across from a white couch
Adv: A white couch across from a green couch

Figure 4.2: Examples from *Ref-Easy* , *Ref-Hard* , and *Ref-Adv* splits. As seen, *Ref-Hard* and *Ref-Adv* have several words in common but differ in their linguistic structure and the target object of interest.

relations between objects) performs worse on *Ref-Adv* even when it performs well on *Ref-Hard* due to exploiting biases in the training data.

Figure 4.2 shows several examples from the *Ref-Easy* , *Ref-Hard* , and *Ref-Adv* splits. We note that *Ref-Adv* expressions are longer on average than *Ref-Easy* and *Ref-Hard* (Figure 6.11 in appendix) and consists of rich and diverse spatial relationships (Figure 4.7 in appendix).

Concurrent to our work, [GAB20] also propose perturbed test splits for several tasks by modifying in-domain examples. In their setup, the original authors of each task create perturbed examples, whereas we use crowdworkers. Closest to our work is from [KHL20] who also use crowdworkers. While we use perturbed examples to evaluate robustness, they also use them to improve robustness (we propose complementary methods to improve robustness §4.5). Moreover, we are primarily concerned with the robustness of models for visual expression recognition task, while [GAB20] and [KHL20] focus on different tasks (e.g., sentiment, natural language inference).

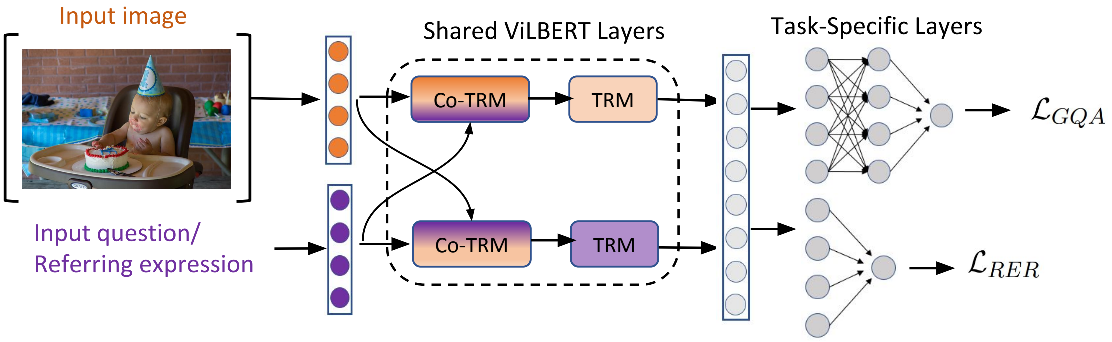


Figure 4.3: Multi-task learning model for referring expression recognition with GQA

4.3.1 Human Performance on *Ref-Easy* , *Ref-Hard* and *Ref-Adv*

We conducted an additional human study (on AMT) to compare the human performance on *Ref-Easy* , *Ref-Hard* and *Ref-Adv* splits. First, we randomly sampled 100 referring expressions from each of the three splits. Each referring expression is then assigned to three AMT workers and are asked to select a bounding box to identify the target object. We considered a sample to be correctly annotated by humans if at least two out of three workers select the ground-truth annotation. Through this evaluation, we obtained human performance on each of the three splits Ref-Easy, Ref-Hard, and Ref-Adv as 98%, 95%, and 96% respectively.

4.4 Diagnosing Referring Expression Recognition models

We evaluate the following models, most of which are designed to exploit linguistic structure.

CMN (Compositional Modular Networks; hu2017modeling,andreas2016neural) grounds expressions using neural modules by decomposing an expression into <subject, relation, object> triples. The subject and object are localized to the objects in the image using a localization module while the relation between them is modeled using a relationship module. The full network learns to jointly decompose the input expression into a triple while also recognizing the target object.

GroundNet [CBM18] is similar to CMN, however it makes use of rich linguistic structure (and correspondingly rich modules) as defined by an external syntactic parser.

MattNet [YLS18] generalizes CMN to flexibly adapt to expressions that cannot be captured by the fixed template of CMN. It introduces new modules and also uses an attention mechanism to weigh modules.

ViLBERT [LBP19a], the state-of-the-art model for referring expression recognition, uses a *pretrain-then-transfer* learning approach to jointly learn visiolinguistic representations from large-scale data and utilizes them to ground expressions. This is the only model that does not explicitly model compositional structure of language, but BERT-like models are shown to capture syntactic structure latently [HM19a].

4.4.1 Results and discussion

We trained on the full training set of *RefCOCOg* and performed hyperparameter tuning on a development set. We used the development and test splits of mao2016generation. Table 4.2 shows the model accuracies on these splits and our proposed datasets. The models are trained to select ground truth bounding box from a set of predefined bounding boxes. We treat a prediction as positive if the predicted bounding box has IoU $>$ 0.5 with the ground truth.

Although the overall performance on the test set seem high, in reality, models excel only at *Ref-Easy* while performing poorly on *Ref-Hard* . The difference in performance between *Ref-Easy* and *Ref-Hard* ranges up to 15%. This indicates that current models do not exploit linguistic structure effectively. When tested on *Ref-Adv* , the performance goes down even further, increasing the gap between *Ref-Easy* and *Ref-Adv* (up to 26%). This suggests that models are relying on reasoning shortcuts found in training than actual understanding. Among the models, GroundNet performs worse, perhaps due to its reliance on rigid structure predicted by an external parser and the mismatches between the predicted structure and spatial relations between objects. ViLBERT achieves the highest performance and is relatively more robust than other models. In the next

Model	*Dev*	*Test*	*Easy*	*Hard*	*Adv*
GroundNet	66.50	65.80	67.11	54.47	42.90
CMN	70.00	69.40	69.55	68.63	49.50
MattNet	79.21	78.51	80.96	65.94	54.64
ViLBERT	83.39	83.63	**85.93**	**72.00**	70.90

Table 4.2: Accuracy of models on *RefCOCOg* standard splits and our splits *Ref-Easy* , *Ref-Hard* and *Ref-Adv* .

section, we propose methods to further increase the robustness of ViLBERT.

4.5 Increasing the robustness of ViLBERT

We extend ViLBERT in two ways, one based on contrastive learning using negative samples, and the other based on multi-task learning on GQA [HM19b], a task that requires linguistic and spatial reasoning on images.

Contrastive learning using negative samples Instead of learning from one single example, contrastive learning aims to learn from multiple examples by comparing one to the other. In order to increase the sensitivity to linguistic structure, we mine negative examples that are close to the current example and learn to jointly minimize the loss on the current (positive) example and maximize the loss on negative examples. We treat the triplets $\big(i, e, b\big)$ in the training set as positive examples, where i, e, b stands for image, expression and ground truth bounding box. For each triplet $\big(i, e, b\big)$, we sample another training example $\big(i', e', b'\big)$, and use it to create two negative samples, defined by $\big(i', e, b'\big)$ and $\big(i, e', b\big)$, i.e., we pair wrong bounding boxes with wrong expressions. For efficiency, we only consider negative pairs from the mini-batch. We modify the batch loss function as follows:

$$\begin{aligned}\mathcal{L}\big(\mathbf{i},\mathbf{e},\mathbf{b}\big)=&\mathbf{F}_{(\mathbf{e},\mathbf{e}')}\Big[\ell\big(\mathbf{i},\mathbf{e},\mathbf{b}\big)-\ell\big(\mathbf{i},\mathbf{e}',\mathbf{b}\big)-\tau\Big]_{+}\\&+\mathbf{F}_{(\mathbf{i},\mathbf{i}')}\Big[\ell\big(\mathbf{i},\mathbf{e},\mathbf{b}\big)-\ell\big(\mathbf{i}',\mathbf{e},\mathbf{b}'\big)-\tau\Big]_{+}\end{aligned}$$

Here $\ell(i,e,b)$ is the cross-entropy loss of ViLBERT, $[x]_{+}$ is the hinge loss defined by $\max\big(0,x\big)$, and τ is the margin parameter. F indicates a function over all batch samples. We define F to be either sum of hinges (Sum-H) or max of hinges (Max-H). While Sum-H takes sum over all negative samples, If batch size is n, for each $\big(i,e,b\big)$, there will be $n-1$ triplets of $\big(i',e,b'\big)$ and $\big(i,e',b\big)$. For $\big(i,e,b\big)$, there will be one $\big(i',e,b'\big)$ and one $\big(i,e',b\big)$. Similar proposals are known to increase the robustness of vision and language problems like visual-semantic embeddings and image description ranking [KSZ14, GSK17, FFK18].

Multi-task Learning (MTL) with GQA In order to increase the sensitivity to linguistic structure, we rely on tasks that require reasoning on linguistic structure and learn to perform them alongside our task. We employ MTL with GQA [HM19b], a compositional visual question answering dataset. Specifically, we use the GQA-Rel split which contains questions that require reasoning on both linguistic structure and spatial relations (e.g., *Is there a boy wearing a red hat standing next to yellow bus?* as opposed to *Is there a boy wearing hat?*). Figure 4.3 depicts the neural architecture. We share several layers between the tasks to enable the model to learn representations useful for both tasks. Each shared layer constitute a co-attention transformer block (Co-TRM; lu2019vilbert) and a transformer block (TRM; vaswani2017attention). While in a transformer, attention is computed using queries and keys from the same modality, in a co-attention transformer they come from different modalities (see cross arrows in Figure 4.3). The shared representations are eventually passed as input to task-specific MLPs. We optimize each task using alternative training [LLS15].

Results and discussion Table 4.3 shows the experimental results on the referring expression recognition task. Although contrastive learning improves the robustness of ViLBERT on *Ref-Adv* (+1.4% and +2.5% for Sum-H and Max-H respectively), it comes at a cost of slight performance

Model	*Dev*	*Test*	*Easy*	*Hard*	*Adv*
ViLBERT (VB)	83.39	83.63	85.93	72.00	70.90
VB+*Sum-H*	81.61	83.00	85.93	70.60	72.30
VB+*Max-H*	82.93	82.70	86.58	70.46	73.35
VB+*MTL (GQA)*	83.45	84.30	86.23	**73.79**	73.92

Table 4.3: Accuracy of enhanced ViLBERT models.

drop on the full test (likely due to sacrificing biases shared between training and test sets). Whereas MTL improves the robustness on all sets showing that multi-task learning helps (we observe 2.3% increase on GQA §4.7.5.2). Moreover, the performance of MTL on *Ref-Hard* and *Ref-Adv* are similar, suggesting that the model generalizes to unseen data distribution. Figure 5.3 shows qualitative examples comparing MTL predictions on *Ref-Hard* and *Ref-Adv* parallel examples. These suggest that the MTL model is sensitive to linguistic structure. However, there is still ample room for improvement indicated by the gap between *Ref-Easy* and *Ref-Hard* (12.4%).

4.6 Summary

In this Chapter, we show that current datasets and models for visual referring expressions fail to make effective use of linguistic structure. Although our proposed models are slightly more robust than existing models, there is still significant scope for improvement. We hope that *Ref-Hard* and *Ref-Adv* will foster more research in this area.

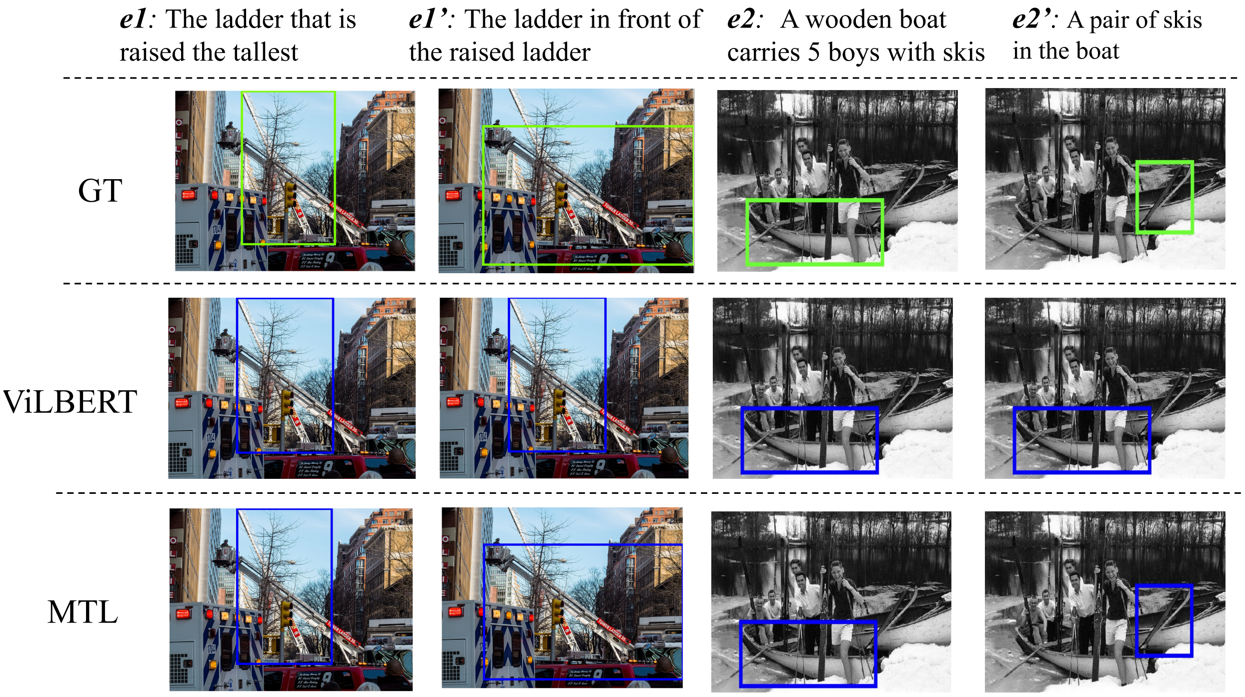

e1: The ladder that is raised the tallest
e1': The ladder in front of the raised ladder
e2: A wooden boat carries 5 boys with skis
e2': A pair of skis in the boat
GT
ViLBERT
MTL

Figure 4.4: Predictions of ViLBERT and MTL model (GT denotes ground-truth). $e1'$ and $e2'$ are adversarial expressions of $e1$ and $e2$ respectively.

4.7 Appendix

In this supplementary material, we begin by providing more details on *RefCOCOg* dataset to supplement Section 4.2 of the Chapter 4. We then provide *Ref-Adv* annotation details, statistics, analysis, and random examples, to supplement Section 4.3 of the Chapter 4. Finally, we provide details of our models (initialization & training, hyper-parameters) and show additional results to supplement Section 4.5 of the Chapter 4.

4.7.1 *RefCOCOg* vs Other Referring Expressions Datasets

RefCOCO , *RefCOCO+* [KOM14] and *RefCOCOg* (Google-RefCOCO; mao2016generation) are three commonly studied visual referring expression recognition datasets for real images. All the three data sets are built on top of MSCOCO dataset [LMB14a] which contains more than

300,000 images, with 80 categories of objects. *RefCOCO* , *RefCOCO+* were collected using online interactive game. *RefCOCO* dataset is more biased towards person category. *RefCOCO+* does not allow the use of location words in the expressions, and therefore contains very few spatial relationships. *RefCOCOg* was not collected in an interactive setting and therefore contains longer expressions.

For our adversarial analysis, we chose *RefCOCOg* for the following three important reasons: Firstly, expressions are longer (by 2.5 times on average) in *RefCOCOg* and therefore contains more spatial relationships compared to other two datasets. Secondly, *RefCOCOg* contains at least 2 to 4 instances of the same object type within the same image referred by an expression. This makes the dataset more robust, and indirectly puts higher importance on grounding spatial relationships in finding the target object. Finally, as shown in Table 4.4, *RefCOCO* and *RefCOCO+* are highly skewed towards *Person* object category ($\approx$ 50%) whereas *RefCOCOg* is relatively less skewed ($\approx$ 36%), more diverse, and less biased.

4.7.2 Importance of Linguistic Structure

[CMB18] observed that existing models for *RefCOCOg* are relying heavily on the biases in the data than on linguistic structure. We perform extensive experiments to get more detailed insights into this observation. Specifically, we distort linguistic structure of referring expressions in the *RefCOCOg* test split and evaluate the SOTA models that are trained on original undistorted *RefCOCOg* training split. Similar to [CMB18], we distort the test split using two methods: (a) randomly shuffle words in a referring expression, and (b) delete all the words in the expression except for nouns and adjectives. Table 4.5 shows accuracies for the models with (column 3 and 4) and without (column 2) distorted referring expressions. Except for the ViLBERT model[LBP19a], the drop in accuracy is not significant indicating that spatial relations are ignored in grounding the referring expression.

Using the relatively robust ViLBERT model, we repeat this analysis on our splits *Ref-Easy* , *Ref-Hard* and *Ref-Adv* . We randomly sampled 1500 expressions from each of these splits and

	RefCOCO	*RefCOCO+*	*RefCOCOg*
Outdoor	0.89%	0.88%	1.65%
Food	10.16%	10.07%	8.10%
Indoor	3.10%	3.09%	2.59%
Appliance	0.67%	0.68%	1.03%
Kitchen	3.95%	3.95%	5.40%
Accessory	2.33%	2.33%	2.85%
Person	49.50%	49.70%	37.02%
Animal	13.26%	13.27%	15.05%
Vehicle	7.23%	7.22%	10.71%
Sports	0.73%	0.74%	1.91%
Electronic	1.94%	1.95%	2.56%
Furniture	6.14%	6.12%	11.09%

Table 4.4: Distribution of object categories in *RefCOCO* , *RefCOCO+* , and *RefCOCOg* datasets.

then compare performance of ViLBERT on these three sets. As shown in Table 4.6, we find a large difference in model's accuracy on *Ref-Hard* and *Ref-Adv* . This clearly indicates that grounding expressions in both of these splits require linguistic and spatial reasoning.

4.7.3 *Ref-Adv* Annotation

We construct *Ref-Adv* by using all the 9602 referring expressions from *RefCOCOg* test data split. As shown in Figure 4.5, we follow a three stage approach to collect these new samples:

Stage 1: For every referring expression in *RefCOCOg* test split, we perturb its linguistic structure by shuffling the word order randomly. We show each of these perturbed expression along with

Model	Original	Shuf	N+J
CMN [HRA17]	69.4	66.4	67.4
GroundNet [CBM18]	65.8	57.6	62.8
MattNet [YLS18]	78.5	75.3	76.1
ViLBERT [LBP19a]	83.6	71.4	73.6

Table 4.5: *RefCOCOg* test accuracies of SOTA models on (a) original undistorted split, (b) after randomly shuffling words (Shuf) in the referring expression, and (c) after deleting all the words except for nouns and adjectives (N+J). ViLBERT is relatively more robust than other baselines.

images and all object bounding boxes to five qualified Amazon Mechanical Turk (AMT) workers and ask them to identify the ground-truth bounding box for the shuffled referring expression. We hired workers from US and Canada with approval rates higher than 98% and more than 1000 accepted HITs. At the beginning of the annotation, we ask the turkers to go through a familiarization phase where they become familiar with the task. We consider all the image and expression pairs for which at least 3 out of 5 annotators **failed to locate** the object correctly (with IoU < 0.5) as hard samples (***Ref-Hard***). We refer to the image-expressions for which at least 3 out of 5 annotators were **able to localize** the object correctly as easy samples (***Ref-Easy***). On average, we found that humans failed to localize the objects correctly in 17% of the expressions.

Stage 2: We take *Ref-Hard* images and ask turkers to generate adversarial expressions such that the target object is different from the original object. More concretely, for each of the hard samples, we identify the most confused image regions among human annotators as the target objects in stage 1. For each of these target objects, we then ask three turkers to write a referring expression while retaining at least three content words (nouns and adjectives) in the original referring expression. This generates adversarial expressions for the original ground-truth *Ref-Hard* referring expressions.

Test	Original	Shuf	N+J
Ref-Easy	86.40	75.06	76.00
Ref-Hard	72.73	51.13	56.60
Ref-Adv	71.08	50.23	57.40

Table 4.6: *Ref-Easy* , *Ref-Hard* , and *Ref-Adv* test accuracies of ViLBERT on (a) original undistorted split, (b) after randomly shuffling words (Shuf) in the referring expression, and (c) after deleting all the words except for nouns and adjectives (N+J).

Referring Expressions	3704
Unique Images	976
Vocabulary	2319
Avg. Length of Expression	11.4

Table 4.7: *Ref-Adv* Statistics

Stage 3: We filter out the noisy adversarial expressions generated in stage 2 by following a validation routine used in the generation of *RefCOCOg* dataset. We ask three additional AMT workers to select a bounding box to identify the target object in the adversarial expression and then remove the noisy samples for which the inter-annotator agreement among workers is low. The samples with at least 2 out of 3 annotators achieving IoU $>$ 0.5 will be added to *Ref-Adv* dataset.

4.7.4 Dataset Analysis, Comparison, and Visualization

In Table 4.7 we summarize the size and complexity of our *Ref-Adv* split. Figure 6.11 shows expression length distribution of *Ref-Easy* , *Ref-Hard* , and *Ref-Adv* . It should be noted that *Ref-Adv* expressions are longer on average than *Ref-Easy* and *Ref-Hard* . Distribution of object categories in *Ref-Easy* , *Ref-Hard* and *Ref-Adv* is shown in Table 4.8. In comparison to *Ref-Easy* and *Ref-Hard* ,

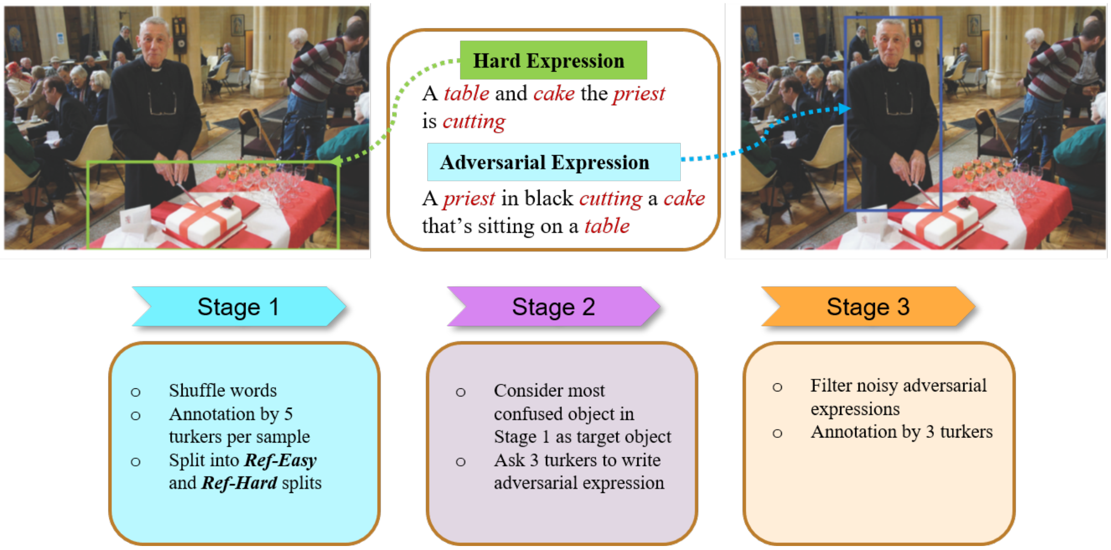

Hard Expression
A table and cake the priest is cutting
Adversarial Expression
A priest in black cutting a cake that's sitting on a table
Stage 1
Stage 2
Stage 3
Shuffle words
Annotation by 5 turkers per sample
Split into Ref-Easy and Ref-Hard splits
Consider most confused object in Stage 1 as target object
Ask 3 turkers to write adversarial expression
Filter noisy adversarial expressions
Annotation by 3 turkers

Figure 4.5: Overview of our three-stage *Ref-Adv* construction process. Given the image, referring expression, ground-truth bounding boxes for all the samples in *RefCOCOg* test split, we first filter out the hard samples and then construct adversarial expressions using them. Please refer to section 2 for further detail.

Ref-Adv is more balanced and less biased towards `Person` category. Figure 4.7 shows the relative frequency of the most frequent spatial relationships in all the three splits. As we can see, *Ref-Adv* comprises of rich and diverse spatial relationships. In Table 4.2, we show random selection of the *Ref-Easy* , *Ref-Hard* , and *Ref-Adv* splits.

4.7.5 Model and other Experiment Details

4.7.5.1 Datasets

GQA [HM19b] contains 22M questions generated from Visual Genome [KZG17] scene graphs. However, in our our multi-task training (MTL), we leverage only 1.42M questions that require reasoning on both linguistic structure and spatial relations. We filter these relational questions by applying the following constraint on question types: *type.Semantic*=*'**rel**'*. We also apply this

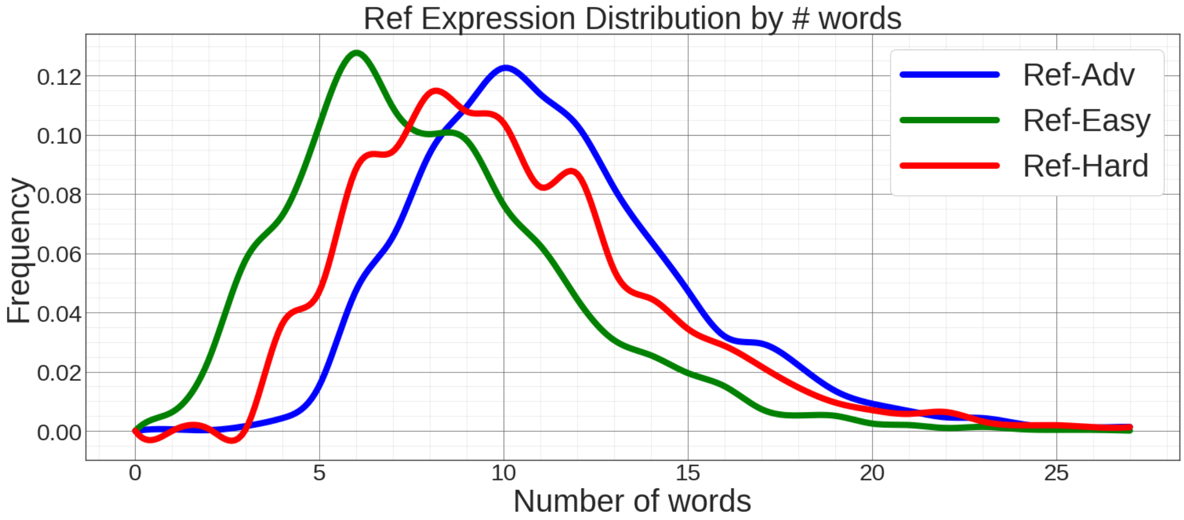


Figure 4.6: Referring expression length distribution for *Ref-Easy* , *Ref-Hard* , *Ref-Adv* datasets.

constraint for filtering the development set. We denote this subset as *GQA-Rel*. We considered GQA-Rel instead of GQA for two reasons: 1) GQA-Rel is a more related task to RefCOCOg; and 2) MTL training with the full GQA set is computationally expensive. For each question in the dataset, there exists a long answer (free-form text) and a short answer (containing one or two words). We only consider the short answers for the questions and treat the unique set of answers as output categories. While the full GQA dataset has 3129 output categories, GQA-Rel contains only 1842 categories.

We follow yu2018mattnet in creating the train (80512 expressions), val (4896 expressions), and test (9602 expressions) splits of ***RefCOCOg*** . For all our experiments in this paper, we directly use the ground-truth bounding box proposals.

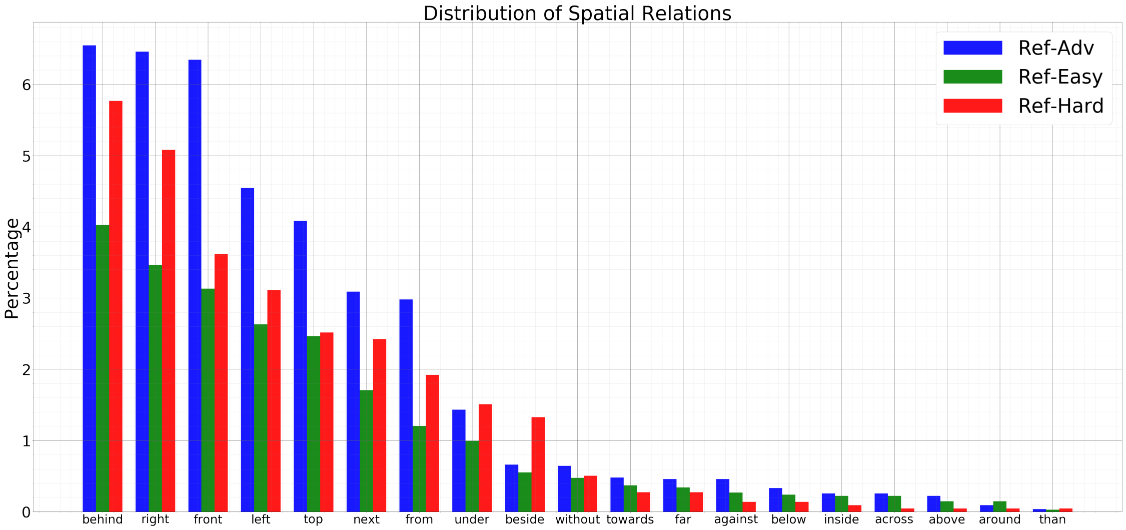


Figure 4.7: Relative frequency of the most frequent spatial relationships in *Ref-Easy* , *Ref-Hard* , and *Ref-Adv*

4.7.5.2 Training

ViLBERT Pre-training We used pre-trained ViLBERT model that is trained on 3.3 million image-caption pairs from Conceptual Captions dataset [SDG18a].[‡]

Single-Task Fine-tuning on *RefCOCOg* In order to fine-tune the baseline ViLBERT [LBP19a] model on *RefCOCOg* dataset, we pass the ViLBERT visual representation for each bounding box into a linear layer to predict a matching score (similar to RefCOCO+ training in lu2019vilbert). We calculate accuracy using IoU metric (prediction is correct if IoU(predicted_region, ground-truth region) $>$ 0.5). We use a binary cross-entropy loss and train the model for a maximum of 25 epochs. We use early-stopping based on the validation performance. We use an initial learning rate of 4e-5 and use a linear decay learning rate schedule with warm up. We train on 8 Tesla V100 GPUs with a total batch size of 512.

[‡] ViLBERT 8-Layer model at the link `https://github.com/jiasenlu/vilbert_beta`

	Ref-Easy	*Ref-Hard*	*Ref-Adv*
	8034 samples	1568 samples	3704 samples
Outdoor	1.21%	1.90%	1.97%
Food	7.94%	9.80%	9.63%
Indoor	2.81%	2.83%	2.76%
Appliance	0.80%	1.07%	1.11%
Kitchen	4.52%	5.73%	5.77%
Accessory	3.20%	5.44%	5.29%
Person	37.26%	20.88%	21.01%
Animal	15.95%	13.92%	13.90%
Vehicle	10.91%	10.40%	10.26%
Sports	1.45%	5.04%	5.13%
Electronic	2.62%	3.20%	3.31%
Furniture	11.28%	19.73%	19.83%

Table 4.8: Distribution of object categories in *Ref-Easy* , *Ref-Hard* , and *Ref-Adv* splits.

Negative Mining We used a batch size of 512 and randomly sample negatives from the mini-batch for computational efficiency. We sampled 64 negatives from each batch for both Sum of Hinges and Max of Hinges losses. We fine-tune the margin parameters based on development split. We train the model for a maximum of 25 epochs. We use early-stopping based on the validation performance. We use an initial learning rate of 4e-5 and use a linear decay learning rate schedule with warm up. We train on 8 Tesla V100 GPUs with a total batch size of 512.

Multi-Task Learning (MTL) with GQA-Rel The multi-task learning architecture is shown in Figure 4.3 in the main paper. The shared layers constitute transformer blocks (TRM) and co-attentional transformer layers (Co-TRM) in ViLBERT [LBP19a]. The task-specific layer for GQA task is a two-layer MLP and we treat it as a multi-class classification task and the task-specific layer

Split	Before MTL	After MTL
GQA-Rel Dev	53.7%	56.0%
GQA Dev	40.24%	42.1%
GQA Test	36.64%	39.2%

Table 4.9: Performance on GQA-Rel Dev, GQA-Dev and GQA-Test splits *before* and *after* MTL training with *RefCOCOg* (Note: MTL training for all the three rows is performed using GQA-Rel and *RefCOCOg*).

for RER is a linear layer that predicts a matching score for each of the image regions given an input referring expression. The weights for the task-specific layers are randomly initialized, whereas the shared layers are initialized with weights pre-trained on 3.3 million image-caption pairs from Conceptual Captions dataset [SDG18a]. We use a binary cross-entropy loss for both tasks. Similar to luong2015multi, during training, we optimize each task alternatively in mini-batches based on a mixing ratio. We use early-stopping based on the validation performance. We use an initial learning rate of 4e-5 for *RefCOCOg* and 2e-5 for GQA, and use a linear decay learning rate schedule with warm up. We train on 4 RTX 2080 GPUs with a total batch size of 256.

GQA MTL Results Table 3 in the main paper showed that MTL training with GQA-Rel significantly improved the performance of model on *Ref-Hard* and *Ref-Adv* splits. In addition, we also observed a significant improvement in GQA-Rel development, GQA development and test splits as shown in the Table 4.9.

4.7.5.3 Additional Experiments

In this subsection, we present results of additional experiments using transfer learning (TL) and multi-task learning (MTL) with ViLBERT on VQA, GQA, and GQA-Rel tasks. As shown in Table 4.10, TL with VQA showed slight improvement. However, TL with GQA, TL with GQA-Rel,

ViLBERT	*Ref-Dev*	*Ref-Test*	*Ref-Adv*
Without TL and MTL	83.39	83.63	70.90
TL with VQA	82.26	84.14	72.96
TL with GQA	80.60	82.08	70.41
TL with GQA-Rel	81.05	83.12	70.78
MTL with VQA	81.20	82.10	70.82
MTL with GQA-Rel	**83.45**	**84.30**	**73.92**

Table 4.10: Comparing ViLBERT's Multi-task Learning (MTL) with Transfer Learning (TL) experiments. *Ref-Dev* and *Ref-Test* correspond to: *RefCOCOg-Dev* and *RefCOCOg-Test* splits respectively.

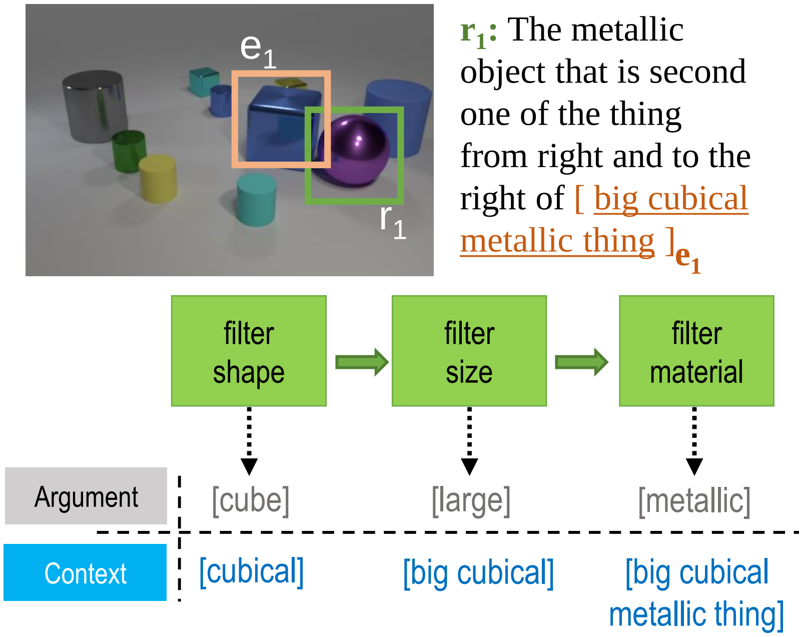
and MTL with VQA did not show any improvements [§].

[§] We could not perform MTL with GQA as it requires large number of computational resources.

CHAPTER 5

Improving Robustness and Faithfulness of Neural Module Networks

In this Chapter, we show that the state-of-the-art end-to-end modular network (NMNs) implementations [SBG20, AGA20a, AJC21, AGW21] - although provide high model interpretability with their transparent, hierarchical and semantically motivated architecture - require a large amount of training data and are less effective in generalizing to unseen but known language constructs. For example, NMNs fail to understand new concepts such as "*yellow sphere to the left*" that are constructed using a combinations of known concepts from train data such as "*blue sphere*", "*yellow cube*", and "*metallic cube to the left*". One of the main reasons for this is that the neural modules in existing works either use a shallow, indirect language guidance [PSV18, HAR17, ASM13] or pre-define the textual inputs in the module instantiation [JHM17b, LLB19], ignoring the rich correlations among the visual inputs and the relevant context from the textual inputs. For example, the neural module that filters based on the object size, "`filter_size(smallest)`", needs to localize a tiny sphere or a medium-sized sphere in the image depending on the object relationships in the expression (e.g. "*the smallest thing among the spheres*" vs. "*the metallic sphere smaller than all the large cylinders*") and the different sizes of spheres and cylinders available in its visual input. We believe that explicitly conditioning the neural modules on the joint textual and visual context helps in inferring robust visiolinguistic relationships which further enhances the compositional reasoning skills. In this Chapter, we propose several extensions to modular networks that mitigate bias in the training and improve robustness and faithfulness of model.



$\mathbf{r_1}$: The metallic object that is second one of the thing from right and to the right of [big cubical metallic thing]$_{\mathbf{e_1}}$

Figure 5.1: An example from the CLEVR-Ref+ dataset. In addition to passing textual inputs (arguments) `cubical`, `large` and `metallic` to neural modules, we also provide them with the relevant neighborhood of arguments as context (highlighted in blue).

5.1 Introduction

Recently, neural module networks (NMN; andreas2016neural,hu2017modeling,liu2019clevrref) have been gaining popularity as a promising approach for solving this task. Briefly, NMN models use an explicit modular reasoning process where a program generator first analyzes the input referring expression and predicts a sequence of learnable *neural modules* (e.g. `count`, `filter`, `compare`). Next, an execution engine dynamically assembles these modules to predict the target object in the image. Such a module based hierarchical reasoning process helps NMNs in providing high model interpretability and therefore facilitates in improving overall trust in the model [ARD16b, AWZ20].

Although achieving promising results, existing NMN models primarily focused on designing module architectures with textual inputs directly hard-coded in the module instantiation [JHM17b, LLB19]. For example, processing the textual inputs '*red*' and '*blue*' require the instantiation of two different modules `filter_color[red]` and `filter_color[blue]`. However, such a design demands a large number of learnable modules (and network parameters) and they cannot share weights for similar contextual textual inputs (e.g. '*dark cube*' vs. '*black cube*', '*shiny cylinder*' vs. '*metallic cylinder*'). Lack of these contextual signals leads to poor generalization performance on unseen but known language contexts [LB17, BVO19].

Moreover, in the prior implementations of NMN such as IEP-Ref [JHM17b, LLB19], the modules in execution engine are not conditioned on the surrounding context of their textual input in the expression. This is problematic as the modules are not given the opportunity to watch the neighborhood of textual input that helps in extracting the informative visiolinguistic context from the module's visual input. For example, the module `filter_color[dark]` needs to pick a black colored cube or a red-colored cube depending on the neighborhood context in the expression (e.g. "*the dark thing that is hardly visible*" vs. "*the dark thing among the red cubes*") and the type of cubes available in its visual input. Few implementations of NMN such as FiLM [PSV18] and N2NMN [HAR17] parametrize the surrounding context of their textual input. However, the visiolinguistic context in these modules is rather shallow as they cannot jointly co-attend over potential objects of interest directly from the visual input and textual inputs.

In this Chapter, we address the aforementioned issues and evaluate the impact of contextual signals in improving the performance of NMN models. First, we address the problem of hard-coded language inputs by parameterizing the module arguments (Figure 5.1), i.e., for example, we treat "`filter_size`" module as parameterized by textual input "*large*" instead of as a standalone function "`filter_size[large]`" (§5.3). We show that module parametrization reduces the total number of learnable modules by 75% without affecting the performance of NMNs.

Second, we use the ground-truth annotations in CLEVR-Ref+ [LLB19], a challenging synthetic referring expression dataset, to show the evidence that providing the relevant neighborhood context of the textual input to the neural module (see Figure 5.1) is beneficial for improving the model's grounding performance (§5.4.1). We next propose a contextualization method to learn to select the most relevant neighborhood context by jointly co-attending on visual and textual inputs, eliminating the need for ground-truth contextual information (§5.4.2).

Our experimental results show that our approach is effective in capturing visiolinguistic relations and contextual dependencies, especially when the textual inputs are long, and has complex linguistic structures. We demonstrate that our proposed method significantly improves the performance of NMN (§5.5.4) in grounding visual referring expressions. Specifically, on CLEVR-Ref+ benchmark, we outperform competing NMN approaches such as IEP-Ref, FiLM and N2NMN by as much as +8.1% accuracy on single-referent split (S-Ref) and +4.3% on full-referent split (F-Ref). Additionally, we also test our approach on CLOSURE [BVO19] and NLVR2 [SZZ18] benchmarks. CLOSURE is a VQA benchmark consisting of CLEVR-like questions with emphasis on simple and complex referring expressions. NLVR2 is a language grounding task where the goal is to determine whether an expression is true based on two paired real images. Our approach significantly outperforms the existing NMN approaches with +11.2% and +1.7% improvements in accuracy on CLOSURE and NLVR2 respectively.

We further evaluate the impact of our contextualization by constructing a set of contrasting perturbations around CLEVR-Ref+ test instances [GAB20], and call our new dataset CC-Ref+ (§5.5.6). We significantly outperform the state-of-the-art models by as much as +10.4% absolute accuracy on CC-Ref+.

5.2 Related Work

Referring Expression Recognition. Visual referring expression recognition (REF) is the task of identifying the object in an image that is referred to by a natural language expression [MHT16, KOM14]. Datasets containing real images and expressions such as RefCOCO+ [KOM14] and RefCOCOg [MHT16] have been proposed to evaluate the progress on this task. Multi-modal transformers [LBP19a, LYY19, TB19], using pretrain-then-transfer approach, have shown superior performance on these datasets. However, these models fail to learn robust visio-linguistic contextual representations and are shown to exploit the imbalanced distribution in the train and test splits [AGA20a, CBM18]. Recently, CLEVR-Ref+ [LLB19] has been introduced as a synthetic diagnostic benchmark that allows control over dataset bias. There are nearly 0.8M referring expressions of which 32% of expressions refer to only a single object (Single-referent) and 68% refer to more than one object (Multi-referent). In this Chapter, we refer to the full dataset as F-Ref and the single-referent subset as S-Ref. Module network [LLB19, JHM17a, ARD16b] based architectures achieved new state-of-the-art performance on this dataset.

Neural Module Networks. Neural module networks (NMNs) learn to parse textual expressions as executable programs composed of learnable *neural modules* [ARD16b, JHM17a, JHM17b, HAR17]. Each of these modules are specialized to compute basic reasoning tasks and can be assembled to perform complex and compositional reasoning. [ARD16b] used dependency trees [ZZC13] to generate the execution layouts. [ARD16a] proposed dynamic NMNs that learns and adapts the structure of the execution layouts to the question. [JHM17b] proposed homogeneous (IEP) and generic neural modules, unlike fixed and hand-crafted neural module, in which the semantics of each neural module is learnt during training. IEP model achieves promising performance on CLEVR dataset. [LLB19] proposed IEP-Ref by extending IEP model to CLEVR-Ref+ dataset and outperformed all the prior works. Although, compositional by design, the visiolinguistic context in these modules is rather shallow and fail to ground novel combinations of known linguistic constructs [BVO19]. The major difference between our work and these prior works of NMN is that we explicitly parametrize and contextualize the neural modules by jointly attending over the visual and textual inputs.

5.3 Module Parameterization in NMN

We propose parametrization as the first step to enable weight sharing and exploiting associations between similar textual contexts. Specifically, we evaluate the effectiveness of parameterizing module textual inputs using IEP-Ref [LLB19] as the baseline NMN implementation. IEP-Ref,

	Modules
Unary	*Filter Shape*, *Filter Color*, *Filter Material*, *Filter Visible*, *Filter Size*, *Filter Ordinal*, *Unique*, *Relate*, *Same Size*, *Same Shape*, *Same Color*, *Same Material*, *Scene*
Binary	*Intersect*, *Union*

Table 5.1: Modules in Parameterized IEP-Ref

a NMN solution based on IEP [JHM17b], is the current state-of-the-art model on CLEVR-Ref+ dataset.[*] As shown Figure 5.2(a), the neural modules in IEP-Ref are represented using a standard Residual Convolution Block (RCB). Formally, each RCB module (f_n) of arity n receives n feature maps ($\mathbf{F_i}$) of shape $128 \times 20 \times 20$ and outputs a same-sized tensor $f_o = f_n(\mathbf{F_1}, \mathbf{F_2}, ..., \mathbf{F_n})$.

We parameterize each RCB module m as follows: (a) we feed all the words in the textual input e_m into an LSTM; (b) The last hidden state of LSTM h_t is then used to perform element-wise multiplication with the output of the first convolution layer in the RCB block to produce joint representation c_m of module's textual input (e_m) and visual input (v_m), which is then passed to ReLU function (see Appendix):

$$\begin{aligned} \mathbf{h_t} &= \text{LSTM}\left(e_{m,t}, \mathbf{h_{t-1}}\right), \\ \mathbf{c_m} &= \text{conv}\left(v_m\right) \odot \mathbf{h_t}. \end{aligned} \tag{5.1}$$

Table 5.2 shows the count of distinct modules and the model performance before and after parameterizing the RCB modules (i.e. IEP-Ref vs P-Ref). As we can see, there are total 60 distinct modules in IEP-Ref. After parameterization, the distinct number of modules reduce by 75% (i.e., 15 distinct modules) without any drop in the model performance. Table 5.1 presents the list of all the 15 modules in our parameterized NMN model.

[*] We used the IEP-Ref implementation provided at the link `https://github.com/ruotianluo/iep-ref`

	#modules	#param. per module	F-Dev	F-Test	S-Dev	S-Test
IEP-Ref (18K programs)	60	442,752	80.54	78.20	49.89	51.50
P-Ref	15	574,336	**81.23**	**78.31**	**51.60**	**51.57**

Table 5.2: Count of modules, parameters and performance of IEP-Ref and parameterized model (P-Ref).

In addition to evaluating the model performance on the full CLEVR-Ref+ dev (**F-Dev**) and test (**F-Test**) splits, we also evaluate the model on single-referent (S-Ref) dev (**S-Dev**) and test (**S-Test**) splits.[†] Moreover, although the network parameters of each parameterized module slightly increase due to the additional LSTM unit, since each module in IEP-Ref can have multiple instantiations for the same textual input, we have fewer parameters than IEF-Ref in total (see Sec 5.5.4.1 for more discussion).

5.4 Contextualization in NMN

5.4.1 Using Ground-Truth Annotations

We extend our parameterized model by contextualizing it with the neighborhood context of textual input in the referring expression. Figure 5.1 shows an example. We leverage the ground-truth annotations available in CLEVR-Ref+ to provide neighborhood context for the modules as follows: Let us denote the ground-truth neural modules as m_1, m_2, m_3, ..., m_n for a given input referring expression q. Suppose the modules m_j and m_k are children for the parent module m_i in the ground-truth execution tree. We modify the architecture of each neural module shown where we concatenate the ground-truth arguments of all the children modules m_j and m_k and pass it as the neighborhood

[†] For results in the last two columns of Table 5.2, we trained our model using S-Ref train split.

Model	F-Dev	F-Test	S-Dev	S-Test
P-Ref	81.23	78.31	51.60	51.57
P-Ref + Input Expr.	81.10	77.01	50.88	51.45
P-Ref + GT Neighb.	**82.60**	**80.02**	**55.22**	**54.76**

Table 5.3: Performance of contextualized NMN models.

context to the parent module m_i (see Appendix). We test if this contextualization helps.

As an ablation, we also test the model performance where the entire expression q is provided as neighborhood context for the modules instead of the relevant neighborhood. Table 5.3 shows the results. Using the entire expression as the neighborhood context did not show any improvements in the model performance, perhaps due to the difficulty in searching and extracting relevant context from long CLEVR-like expressions. On the other hand, providing ground-truth neighborhood context shows significant improvement in the performance (1.71% on F-test and 3.19% on S-Test), indicating that model is able to extract informative visiolinguistic clues. Since the ground-truth human annotations are costly and difficult to obtain, we next propose a contextualization method that enables the modules to learn to select the most relevant neighborhood context without requiring ground-truth annotations.

5.4.2 Using Memory-augmented Block

We incorporate a memory-augmented LSTM block [GWD14] in the neural module to guide the attention towards the relevant and informative neighborhood words in the input expression (q). Figure 5.2(b) shows our contextualized module architecture. Our design enhances the module's capacity to exploit the visiolinguistic context between the visual input v_m and the selective set of words that are stored in the memory over multiple timesteps.

The memory M consists of a set of row vectors as memory slots. LSTM (i.e., controller) has

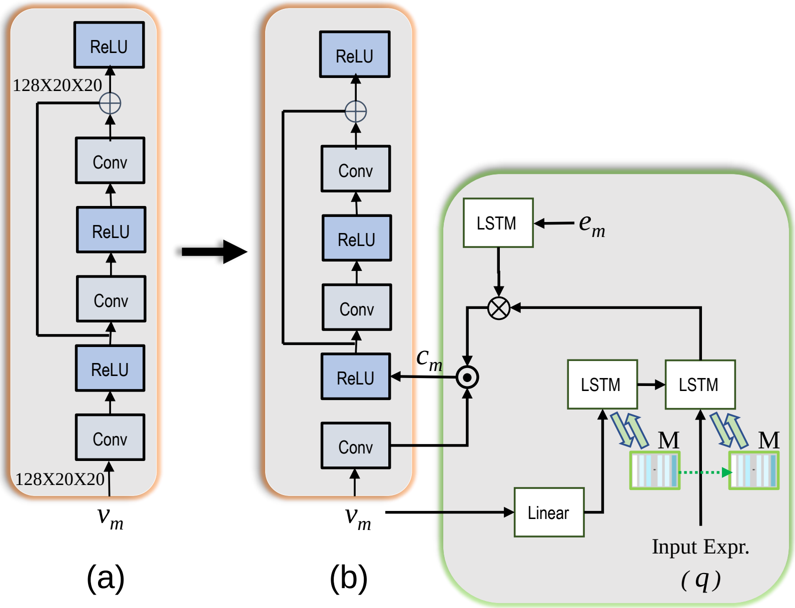


Figure 5.2: (a) Architecture of neural module (m) in IEP-Ref consuming a visual input v_m. $\oplus$ denotes summation. (b) Our proposed contextualized module design using our proposed memory (M) based architecture. $\odot$ and $\otimes$ denote element-wise multiplication and concatenation respectively. e_m is parameterized textual input.

read and write heads into M, which helps in retrieving representations from M or place them into M. In the first time step (t_0), we feed visual input and then in the later time steps textual input is fed. More formally, given a input referring expression q, at each time step (t), LSTM produces a key, $k_{i,t}$, which is either used to retrieve a particular location l from the row M_t or to store in M_t. We feed the referring expression q into LSTM as:

$$\mathbf{h_t} = \text{LSTM}\left(\mathbf{q_t}, \mathbf{h_{t-1}}\right). \tag{5.2}$$

We then compute the cosine similarity measure between h_t and each individual row j in M:

$$K\left(\mathbf{h_t}, \mathbf{M_t}\left(j\right)\right) = \frac{\mathbf{h_t} \cdot \mathbf{M_t}\left(j\right)}{\left\|\mathbf{h_t}\right\| \left\|\mathbf{M_t}\left(j\right)\right\|}. \tag{5.3}$$

A read weight vector w_t is computed using a softmax over the cosine similarity and then a memory row m_t is retrieved. The vectors m_t, h_t are concatenated with the textual input (e_m) and then an

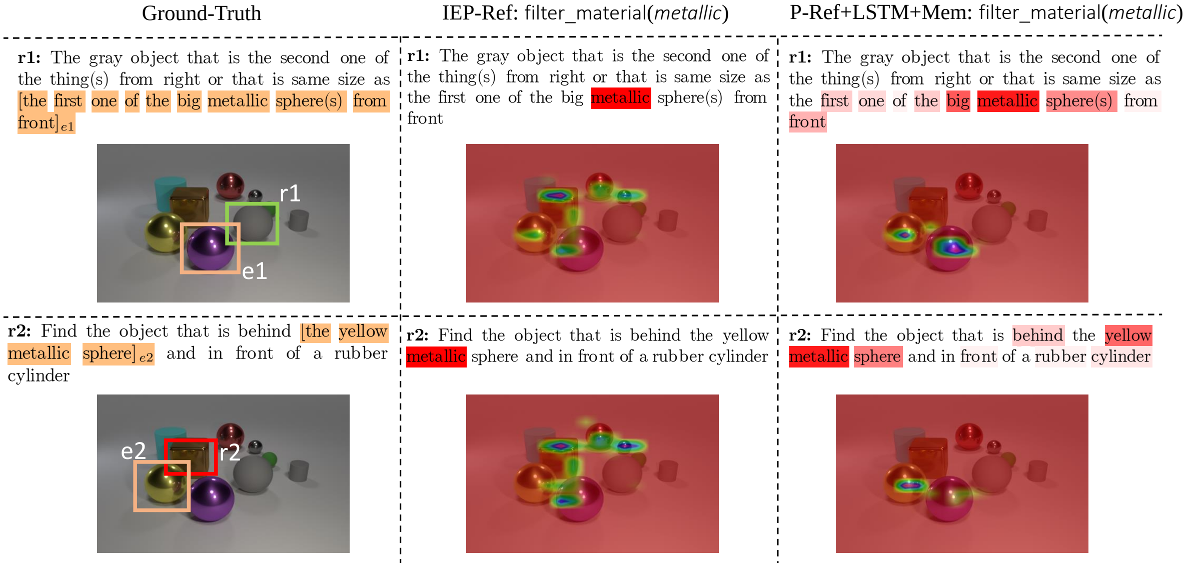

Ground-Truth
IEP-Ref: filter_material(metallic)
P-Ref+LSTM+Mem: filter_material(metallic)
r1: The gray object that is the second one of the thing(s) from right or that is same size as [the first one of the big metallic sphere(s) from front]e1
r1
e1
r1: The gray object that is the second one of the thing(s) from right or that is same size as the first one of the big metallic sphere(s) from front
r1: The gray object that is the second one of the thing(s) from right or that is same size as the first one of the big metallic sphere(s) from front
r2: Find the object that is behind [the yellow metallic sphere]e2 and in front of a rubber cylinder
e2
r2
r2: Find the object that is behind the yellow metallic sphere and in front of a rubber cylinder
r2: Find the object that is behind the yellow metallic sphere and in front of a rubber cylinder

Figure 5.3: Qualitative examples showing the attention heatmaps of `filter_material(metallic)` module outputs trained using IEP-Ref and P-Ref+LSTM+Mem models. $e1$ and $e2$ highlight the metallic objects that are referred in the input expressions $r1$ and $r2$ respectively.

element-wise multiplication is performed with the output of the convolution layer before passing to the ReLU function (see Appendix).

5.5 Experiments

5.5.1 Datasets

We evaluate our approach on F-Ref and S-Ref splits of CLEVR-Ref+ [LLB19]. In addition, we also test our approach on CLOSURE [BVO19] and NLVR2 [SZZ18] benchmarks. CLOSURE is a VQA benchmark, consisting of synthetically generated image and question pairs with emphasis on grounding simple and complex referring expressions. NLVR2 is a language grounding task where the goal is to determine whether an expression is true based on two paired real images. While reporting results on CLOSURE, we train our NMN model using CLEVR [JHM17a] train and val splits.

5.5.2 Baselines

We compare the performance of our approach against the following baselines: (1) **IEP-Ref** [LLB19] is the current state-of-the-art NMN model for CLEVR-Ref+ benchmark which uses explicit program generator and execution engine (PG+EE) to predict the answer; (2) **FiLM** (Feature-wise Linear Modulation) [PSV18] is a NMN model which introduces new layers in the RCB block that learn parameters $\gamma_{i,c}$ and $\beta_{i,c}$ for scaling up or down the CNN activations ($F_{i,c}$) by conditioning on the input referring expression x_i, i.e. $FiLM(\mathbf{F_{i,c}}|\gamma_{i,c}, \beta_{i,c}) = \gamma_{i,c}\mathbf{F_{i,c}} + \beta_{i,c}$; (3) **MAC** [HM19c] is an end-to-end differentiable architecture designed to perform an explicit multi-step reasoning process by decomposing them into a series of attention-based reasoning steps; (4) **VectorNMN** [BVO19] is a direct extension to FiLM that uses vector-valued inputs and outputs for the modules instead of high-capacity 3D tensors; (5) **NS-VQA** [YWG18] uses structural scene representation from input image in addition PG+EE components in IEP-Ref; (7) **N2NMN** uses hand-crafted and parameterized neural modules; (8) **LCGN** [HRD19] uses a graph network where each node represents an object, and is described by a context-aware representation from related objects conditioned on the textual input.

To gain better insight into the relative contribution of the design choices we made, we perform experiment with the following ablated models: (9) **P-Ref+LSTM+Attn** uses attention instead of an external memory block for selecting the neighborhood words in the expression; (10) **P-Ref+Curriculum Learning**: We employ a curriculum training (Platanios$_2$019$)regimetotraintheP-$ $Refmodelinordertoimproveitsperformancewithoutcontextualization(SeeAppendix).$

5.5.3 Implementation Details

The memory matrix in our model discussed in section 5.4.2 consists of 128 rows and 80 columns. The controller is a single layer LSTM network. We use GloVe to obtain the word embedding (dimension = 300) of each word in the textual input. When training, we first train our program generator (PG) and use it as a fixed module for training the execution engine (EE). We use 18K ground-truth programs to train the program generator (PG). We train PG and EE using Adam [KB15] with learning rates 0.0005 and 0.0001, respectively. Note that PG is trained for a maximum of 32,000 iterations, while EE is trained for a maximum of 450,000 iterations. We employ early stopping based on validation set accuracy. We do not find any significant improvements with the joint optimization of PG and EE. We train on one RTX 2080ti GPU with a batch size of 8.

5.5.4 Evaluation

Table 5.4 shows results in comparison with the baselines. We find that our contextual NMN model (P-Ref+LSTM+Mem) significantly outperforms all prior work by large margins. In addition to outperforming NMN baselines such as FiLM, N2NMN, IEP-Ref, we also outperform the non-NMN baselines such as LCGN demonstrating the effectiveness of the introduced memory module in capturing visiolinguistic relations and contextual dependencies from the longer CLEVR-like expressions. Specifically, we achieve +4.3% on F-test and +8.1% on S-Test, compared with the current state-of-the-art NMN model IEP-Ref. Most significant gains on S-Test also suggest the superior generalization skills of our model in learning from fewer training samples.

The ablation results are shown in Table 5.5. As we can see, all the ablative baselines underperform, confirming the importance of our proposed contextualization approach. Specifically the improvements obtained with module contextualization in both IEP-Ref and FiLM demonstrate that our approach can generalize across diverse NMN architectures.

Performance on CLOSURE and NLVR2 benchmarks is shown in Table 5.6. We achieve +11.2% in accuracy on CLOSURE test split compared to the best prior model Vector-NMN, indicating that

Model	F-Dev	F-Test	S-Dev	S-Test
IEP-Ref	80.54	78.20	49.89	51.50
FiLM	76.58	75.71	44.90	46.70
MAC	79.40	77.36	47.20	47.00
Vector-NMN	82.05	77.00	46.72	52.88
NS-VQA	80.08	79.01	48.07	51.66
N2NMN	76.00	75.11	43.62	46.70
LCGN	77.07	74.80	46.88	48.00
P-Ref+LSTM+Mem (**ours**)	**84.82**	**83.05**	**59.76**	**60.04**

Table 5.4: Performance of our memory based contextualized NMN model (P-Ref+LSTM+Mem) and baselines on CLEVR-Ref+.

our model generalizes well to unseen compositions. We also surpass all the existing NMN based models for NLVR2 dataset which has real images unlike synthetic images in CLEVR-Ref+ and CLOSURE.

Figure 5.3 illustrates the qualitative differences of `filter_material(metallic)` module trained using IEP-Ref and our P-Ref+LSTM+Mem model. With IEP-Ref, the model selects all metallic objects from the image, ignoring the context in the expression. On the other hand, our approach correctly locates objects based on their contextual relevance.

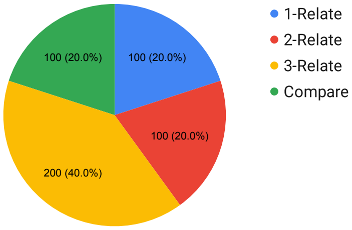
5.5.4.1 Model Parameters

Our proposed model has 3 times fewer parameters than the baseline model IEF-Ref in total (see Table 5.7). More concretely, the baseline IEP-Ref model contains 60 modules and each module consists of 0.44M parameters. That is, total number of parameters in IEP-Ref are 60*0.44M = 26.4M. Similarly, the FiLM baseline, which also does contextualization of inputs, has 60*0.59M

Model	F-Dev	F-Test	S-Dev	S-Test
IEP-Ref	80.54	78.20	49.89	51.50
P-Ref+LSTM+Attn	79.26	78.99	52.68	52.96
P-Ref+LSTM+Mem (**ours**)	**84.82**	**83.05**	**59.76**	**60.04**
FiLM	76.58	75.71	44.90	46.70
FiLM+LSTM+Mem (**ours**)	79.05	80.86	51.10	53.06
P-Ref+CL	81.70	80.32	57.25	56.91
P-Ref+LSTM+Mem+CL	82.16	80.93	57.90	58.14

Table 5.5: **Ablations.** Performance of our model and its ablative baselines on CLEVR-Ref+.

Referring Expressions	500
Unique Images	492
Vocabulary	86
Avg. Length of Expr	20.4



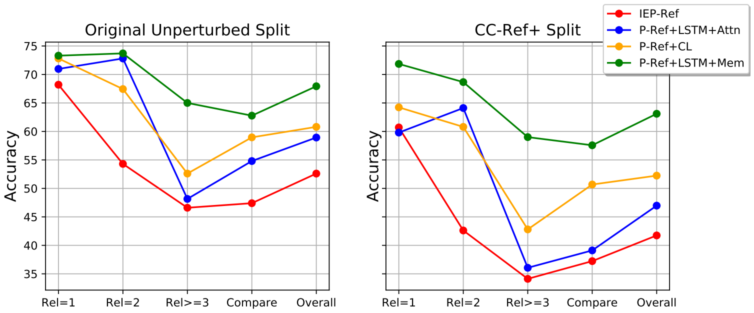


Figure 5.5: Performance of models on randomly drawn 500 original CLEVR-Ref+ test instances and their contrast sets.

Figure 5.4: CC-Ref+ Statistics

= 35.4M parameters. On the other hand, our proposed memory based contextualization of NMN model contains only a maximum of 15 modules and each module has 0.58M parameters. Therefore total number of parameters in our model are 15*0.58M = 8.7M. This is 3 times smaller than IEP-Ref and 4 times smaller than FiLM.

Model	CLOSURE	NLVR2 (Test-P)
IEP-Ref	59.80	N/A
FiLM	58.72	51.10
N2NMN	62.07	52.10
MAC	65.19	51.40
Vector-NMN	64.14	N/A
P-Ref	59.68	N/A
P-Ref+LSTM+Attn	63.13	N/A
P-Ref+LSTM+Mem (**ours**)	**71.22**	N/A
FiLM+LSTM+Mem (**ours**)	69.78	**53.80**

Table 5.6: Performance of our model and NMN baselines on CLOSURE and NLVR2 datasets.

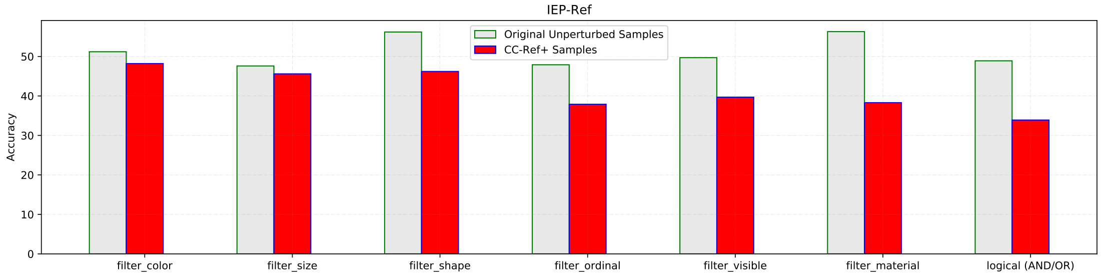


Figure 5.6: Performance of baseline IEP-Ref model on original CLEVR-Ref+ test split and CC-Ref+ samples

5.5.5 The CC-Ref+ Dataset

We further examine the robustness of the models by creating contrast sets (similar to gardner2020evaluating) that help in exposing model brittleness by probing a model's decision boundary local to examples in the test set. Specifically, we follow a three stage approach to collect our contrast set:

Model	#Parameters (per module)
IEP-Ref	442,752
Param. IEP-Ref (P-Ref)	574,336
FiLM	590,720
P-Ref+LSTM+Attn	574,464
P-Ref+CL	574,336
P-Ref+LSTM+Mem (**ours**)	589,597

Table 5.7: Count of parameters for each neural module in the baselines and our proposed NMN models.

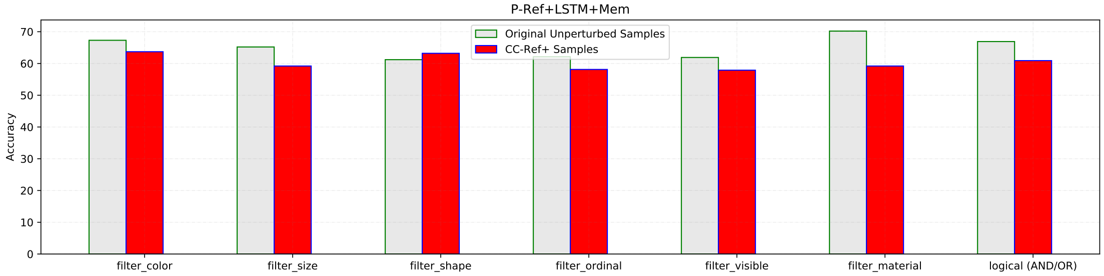


Figure 5.7: Performance of our contextual NMN model (P-Ref+LSTM+Mem) on original CLEVR-Ref+ test split and CC-Ref+ samples

Stage 1: First, we randomly sample 100 single-referent expressions from the test split containing only a single spatial relation (e.g. *The first one of the tiny rubber thing from **left***). We then sample another 100 expressions containing two spatial relations (e.g. *The first one of the thing from **left** that is **behind** the big yellow matte object*). Similarly we sample a third subset of 200 expressions containing 3 or more relations. Finally, we sample 100 expressions containing at least one compare relations (e.g. *Any other tiny object as the **same color** as the big yellow metallic cube*). This constitutes a total of 500 expressions.

Stage 2: We then manually perturb the semantics of various parts of these 500 referring expressions

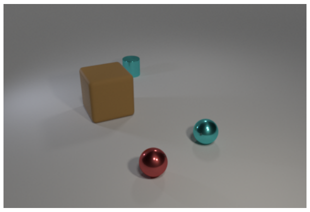
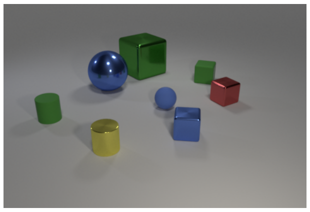
	**Original:** The brown things that are big object(s) or the second one of the small metal thing(s) from left **CC-Ref+:** The cyan things that are big object(s) or the first one of the small metal thing(s) from left
	**Original:** The matte things that are either the sixth one of the tiny thing(s) from right or the fifth one of the thing(s) from front **CC-Ref+:** The matte things that are tiny thing(s) and the second one of the thing(s) from front

Table 5.8: Random examples from CC-Ref+ and their original annotations in CLEVR-Ref+

such that the ground-truth referent object changes. For example, we modify the expression *first one of the tiny rubber thing from left* to *first one of the tiny metallic thing from right*. We call this perturbed test split CC-Ref+. We show random selection of CC-Ref+ examples in Table 5.8.

Stage 3: Finally, we verify and validate the correctness of the new ground-truth annotations using two human annotators. The annotations that are not consistent among the two human annotators are removed and we re-iterate the above three steps until we collect a validated set of 500 contrast samples[‡]. In Figure 5.4, we summarize the size and complexity of our CC-Ref+ split.

[‡] [GAB20] shows that a few hundreds of contrast samples will be sufficient to draw substantiated conclusions about model behavior.

5.5.6 Evaluation on CC-Ref+ Dataset

As shown in Figure 5.5, performance of baseline models drop by $>10\%$ on CC-Ref+ and the models struggle to correctly ground the perturbed samples containing compare relations (e.g. *same_color*) or that containing more than 2 spatial relations (e.g. *front*, *left*) in the expression. Our method shows least drop ($<5\%$) in performance indicating its superiority in grounding expressions with complex linguistic constructs (see Appendix for more detailed analysis). In Figure 5.6 and Figure 5.7, we further analyze the model's performance when one of the object attributes namely, color, size, shape, material, ordinality, and visibility are perturbed in the contrast sets. We found that both IEP-Ref and our model are robust to perturbations in color indicating that this is a relatively easier concept to ground in the images. In contrast to the findings in [LLB19], we see a significant drop by up to 15% in the performance of IEP-Ref on all the other attributes such as shape and visibility. Our proposed approach P-Ref+LSTM+Mem shows relatively low drop in the logical, material and ordinal perturbations, insignificant drops ($< 3\%$) in color, visible perturbations and a slight improvement (+2%) in shape perturbations. This clearly suggests that our approach generalizes well and is robust to contrastive perturbations in the input. The performance gap of P-Ref+LSTM+Mem in logical, ordinal and material perturbations show that these are relatively difficult concepts for the model to learn. We hope that CC-Ref+ dataset will foster more research in this area.

5.6 Another method for Contextualization in NMN

In the previous section, we show the importance of contextualizing modular networks. In this section, we present an alternate method to perform this contextualization. To do this, as shown in Figure 6.1, we replace the standard convolution operations in the neural modules with a novel language-guided adaptive convolution operation, which we call **LG-Conv**. More specifically, the filter weights W of LG-Conv are explicitly multiplied with a spatially varying language-guided kernel G, which allows the module to adaptively co-attend over potential objects of interest from the visual input and textual input by altering the convolution. Although content-adaptive

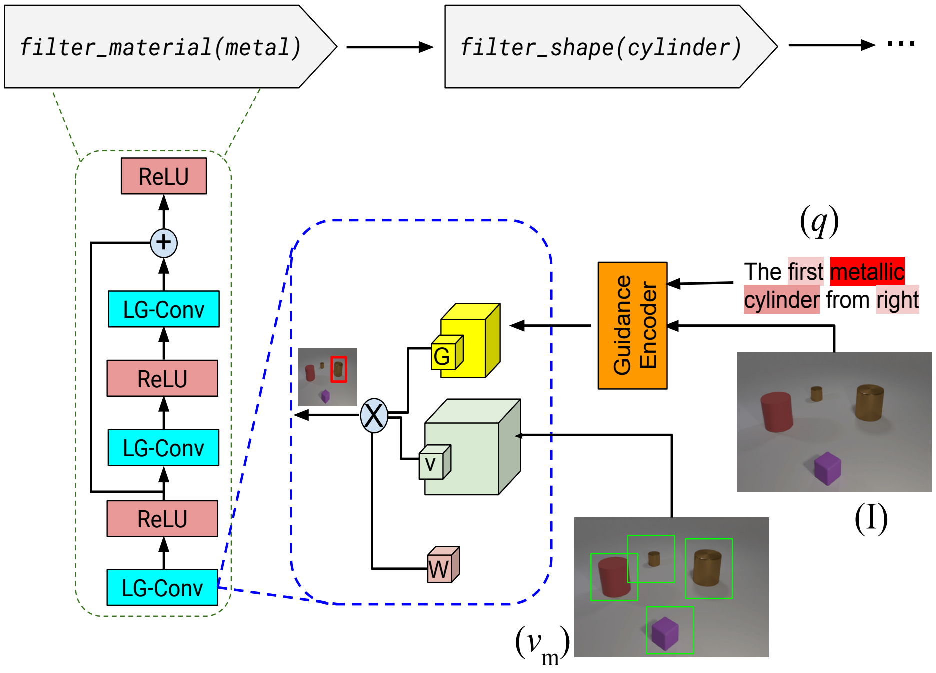


Figure 5.8: An example from the CLEVR-Ref+ dataset. Existing NMN implementations only provide the visual features (v_m) as inputs to the neural modules. In this work, we additionally condition each module on textual expression (q) by replacing the standard convolution layers with content adaptive convolution layers **LG-Conv** which modify the convolution by explicitly multiplying the filter weights (W) with a spatially varying language-guided kernel G. $\bigotimes$ denotes element-wise multiplication and $\bigoplus$ denotes summation.

convolutions [JDT16, DQX17, SJS19] are used in several vision tasks, we are not aware of any prior works that does this filter adaptation using language as guidance. We propose two novel and effective methods namely, bi-salient attentional guidance (BiSAtt) network and co-salient attentional guidance guidance (CoSAtt) network to learn the guidance kernel G from textual and visual inputs.

Problem Setup and Notation. Given an image I and a natural language query q as input, our goal is to develop a NMN model that selects an answer $a \in A$ to the query from a fixed set A of possible answers. We generalize this notation for both VQA and REF tasks; q, a denote question and a natural language answer respectively in VQA, whereas they represent a referring expression

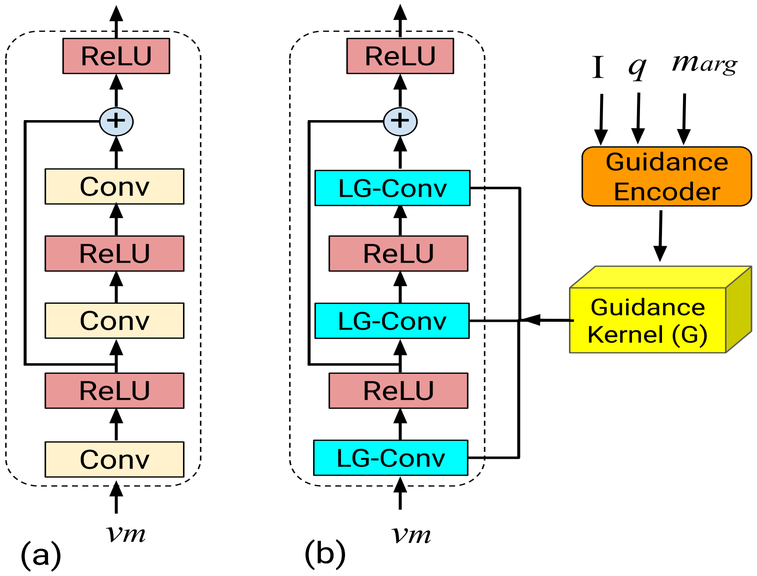


Figure 5.9: (a) Architecture of neural module (m) in existing NMN [JHM17b] consuming a visual input v_m; (b) Our proposed architecture replacing Conv layers with content adaptive convolution layers guided by the input image I, input query q and parameterized textual input m_{arg}.

and a bounding box of the target object respectively in REF. We represent input image I as an ordered sequence of a set of image regions $R = (r_0, r_1, ..., r_N)$ and the query q as the set of words $(w_1, w_2, ..., w_L)$ where w_i is the i-th word, N is the number of image regions extracted from input image I, and L is the total number of the tokens in the input query.

Similar to [JHM17b], we use a two-stage model for generating answer: (1) Program Generation Model $p(z|q; \theta_p)$: where the query is parsed to z representing the reasoning steps required to answer the query, and (2) Program Execution Model $p(a|z, I; \theta_e)$: where the predicted program z is used to assemble a input-specific neural network that is composed from a set of neural modules m and is executed to produce a distribution over answers.

As shown in Figure 5.9(a), the neural modules in current implementations of NMN [JHM17a, LLB19] typically use a standard Residual Convolution Block (RCB), consisting of convolution layers and ReLU activations. Formally, a module (f_n) of arity n receives n feature maps ($\mathbf{F_i}$) of

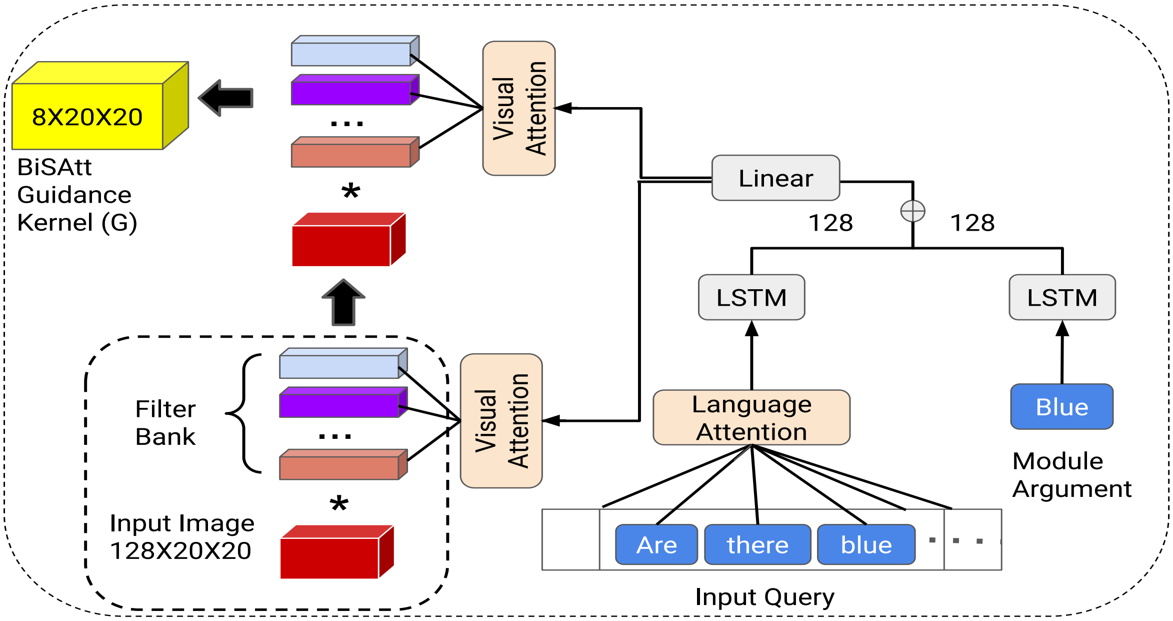


Figure 5.10: Bi-Salient Attentional Guidance Encoder (**BiSAtt**): In BiSAtt Encoder architecture, we first encode text inputs and then use it to learn a set of adaptive weights to linearly combine the basis filters which produces the convolution filters applied on input image.

shape $128 \times 20 \times 20$ and outputs a same-sized tensor $f_o = f_n(\mathbf{F_1}, \mathbf{F_2}, ..., \mathbf{F_n})$.

$$\begin{aligned} h_m &= \text{ReLU}\left(\text{conv}_1\left(F_i\right)\right) \\ f_o &= \text{ReLU}\left(\text{conv}_2\left(h_m\right) \oplus h_m\right) \end{aligned} \tag{5.4}$$

As we can see, these modules [JHM17a, JHM17b] are not explicitly conditioned on the input expression q, and therefore fails to extract robust visiolinguistic relationships. In contrast, as shown in Figure 5.9(b), we explicitly condition the neural modules on q, in addition to visual inputs, by replacing the standard convolution operations in RCB block with a novel adaptive and language-guided convolution operation, which we call LG-Conv. Also, we parameterize the module arguments, i.e., for example, we treat "`filter_material`" module as parameterized by argument (m_{arg}) "*rubber*" instead of as a standalone module "`filter_material[rubber]`". As a result of this parametrization, the number of a distinct set of modules used in the parameterized

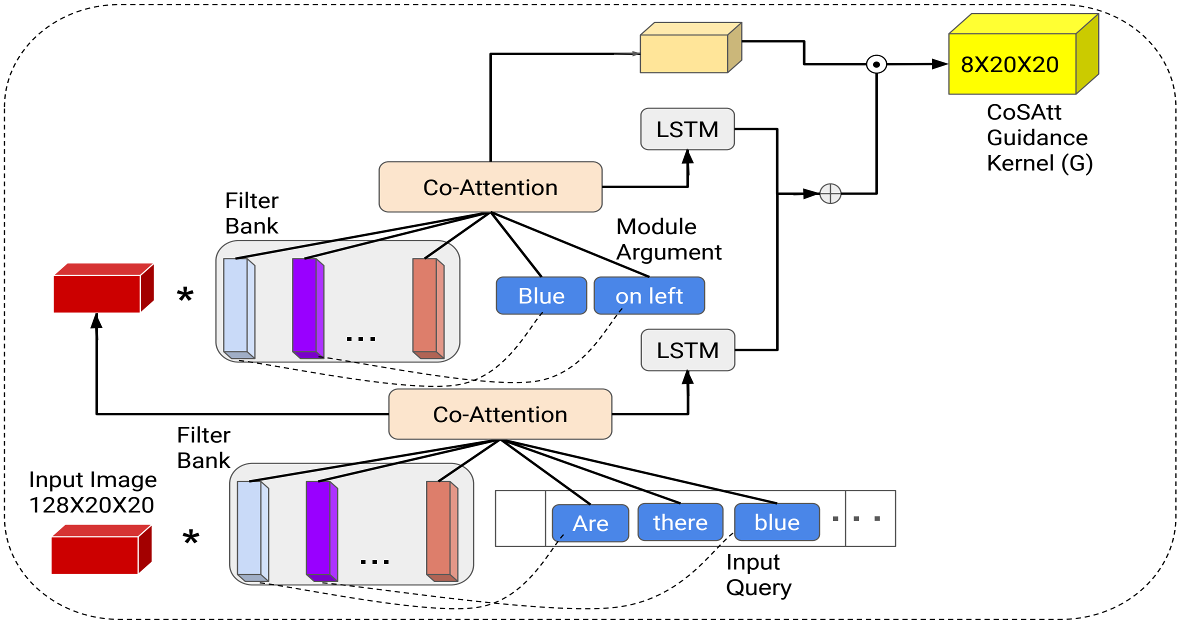


Figure 5.11: Co-Salient Attentional Guidance Encoder (**CoSAtt**): CoSAtt Encoder jointly attends over the input image and text inputs (early fusion) to identify co-salient regions and relationships in visual and language features that are contextually associated with each other.

model reduce by 75%. We condition our LG-Conv layer on both query q and the module argument m_{arg} (See Figure 5.9). In the following, we describe the LG-Conv operation and detail its formal specification.

5.6.1 Language-Guided Convolution

Our high-level goal is to empower the neural modules to learn adapting visiolingustic features from both visual and language inputs. We achieve this by introducing novel LG-Conv layers which allows the module to adaptively co-attend over potential objects of interest from the visual and textual inputs by altering the convolution [JDT16, DQX17, SJS19]. Formally, a standard *conv* layer in the RCB block performing a spatial convolution operation over the n image pixels $P = (p_1, p_2, ...p_n)$ is given as:

$$\mathbf{p'_i} = \sum_{\mathbf{j}\in\mathbf{\Omega(i)}} \mathbf{W}[\mathbf{c_i} - \mathbf{c_j}]\mathbf{p_j} + \mathbf{b} \tag{5.5}$$

where $\mathbf{W}$ denotes the filter weights, $\mathbf{c_i}$ denote the coordinates of the pixels in the image, $\mathbf{b}$ denotes biases, and $\mathbf{\Omega(i)}$ defines a convolution window. This convolution operation, with spatially shared weights, is agnostic to pixel features and independent of language features. As shown in Figure 6.1, we modify this to depend on both pixel features and language features using a spatially varying guidance kernel G as follows:

$$\mathbf{p'_i} = \sum_{\mathbf{j}\in\mathbf{\Omega(i)}} \mathbf{G}(\mathbf{g_i}, \mathbf{g_j})\mathbf{W}[\mathbf{c_i} - \mathbf{c_j}]\mathbf{p_j} + \mathbf{b} \tag{5.6}$$

The spatial convolution W is adapted at each pixel in the visual input using the guidance kernel G. Similar to [SJS19], we represent G using a fixed parametric Gaussian: $G(g_i, g_j) = \exp(-\frac{1}{2}(g_i - g_j)^T(g_i - g_j))$, where g represents guidance features that we learn using the following two methods [§]: (a) **Bi-Salient Attentional Guidance (BiSAtt Encoder)**: We generate spatial guidance features using the architecture shown in Figure 5.10. The input image of dimensions $128 \times 20 \times 20$, the input query q, and the module's parametrized text argument are used in producing the guidance features. Specifically, in BiSAtt architecture, we add visual attention layers over I to generate spatial guidance from non-spatial q. (b) **Co-Salient Attentional Guidance (CoSAtt Encoder)**: Here, we apply a joint attention over I, q, and the module argument to identify co-salient regions and relationships in visual and language features that are contextually associated with each other. The architecture is shown in Figure 5.11. In comparison to BiSAtt, as we show in our experiments, CoSAtt improves the relevance and interaction between objects in the image and the query. For efficient implementation, we use the same learned guidance across all the LG-Conv layers in a RCB block.

[§] We experimented more forms of guidance kernel discussed in [SJS19], but we did not find significant improvements in NMN performance with these other kernels.

As our parametrized model require only a few number of modules, the total number of parameters in our NMN is significantly less compared to the state-of-the-art models, even though the network parameters in our parameterized module slightly increase due to the additional conv and LSTM units in the guidance encoder.

Program Generator. We implement program generator using an attention-based sequence to sequence (seq2seq) model with an encoder-decoder structure [SVL14, JHM17a] to map the input query q into an executable program z. Both the encoder and decoder have two hidden layers with a 256-dim hidden vector. Similar to [JHM17a], we convert the decoded sequence of program functions to syntax trees (in an in-order traversal) in which each node contains a RCB module.

Execution Engine. The execution engine assembles a neural network using the predicted program z by mapping function f at each node in syntax tree to its corresponding neural module. The parent modules in the syntax tree takes the outputs from the child modules. Since we use a homogeneous architecture for designing our modules, the output generated from all modules is of same shape $128 \times 20 \times 20$. We flatten the final feature map before passing it to a multi-layer perceptron classifier, producing a distribution over all possible answers.

Training. During training, we find the optimal module parameters by maximizing the likelihood of the data. We optimize $p(z|q;\theta_p)$ using a policy gradient method.

$$\nabla \mathbf{J}(\theta_{\mathbf{p}}) = \mathbb{E}[\nabla \log \mathbf{p}(\mathbf{z}|\mathbf{q};\theta_{\mathbf{p}}) \cdot \mathbf{r}] \tag{5.7}$$

where r is the reward and the expectation is taken with respect to rollouts of the policy. In order to enforce the network for generating the most accurate predictions, we then train the execution engine directly by maximizing log $p(a|z,I,q;\theta_e)$ with respect to θ_e.

$$\mathbb{E}[\nabla \log \mathbf{p}(\mathbf{z}|\mathbf{q};\theta_{\mathbf{p}}) \cdot \log \mathbf{p}(\mathbf{a}|\mathbf{z},\mathbf{I},\mathbf{q};\theta_{\mathbf{e}})] \tag{5.8}$$

Model	CLEVR-Dev	CLEVR-Test	CLOSURE
IEP-Ref [LLB19]	$98.7^{\pm 0.3}$	$97.1^{\pm 0.2}$	$59.8^{\pm 0.4}$
FiLM [PSV18]	96.2	96.9	58.9
MAC [HM19c]	99.1	98.2	71.6
Vector NMN [BVO19]	98.8	97.6	71.0
NS-VQA [YWG18]	**99.2**	**99.4**	76.4
LCGN [HRD19]	NA	NA	NA
ViLBERT [LBP19b]	95.3	93.0	51.2
Visual BERT [LBP19b]	96.0	92.8	50.6
Ours (with BiSAtt)	$98.9^{\pm 0.2}$	$99.2^{\pm 0.1}$	$86.1^{\pm 0.1}$
Ours (with CoSAtt)	$98.9^{\pm 0.1}$	$99.2^{\pm 0.1}$	$\mathbf{88.0^{\pm 0.2}}$

Table 5.9: Performance of our approach and baselines on CLEVR, CLOSURE benchmarks.

5.6.2 Experiments

Similar to [JHM17a], we use 18K ground-truth programs to train the program generator (PG). We train PG and the execution engine using Adam [KB15] with learning rates 0.0005 and 0.0001, respectively. Our PG is trained for a maximum of 32K iterations, while EE is trained for a maximum of 450K iterations. We employ early stopping based on validation set accuracy. While reporting accuracies on S-Ref test split, we use the model trained on S-Ref train split. We repeat the experiment 5 times on each benchmark and report the mean/variance on each of them.

5.6.3 Evaluation

Table 5.9 and Table 5.10 show results in comparison with the baselines. We find that our model outperforms all prior work on CLOSURE, and CLEVR-Ref+ benchmarks, while showing on-par

Model	S-D	S-T	F-D	F-T
IEP-Ref	$49.8^{\pm 0.1}$	$51.5^{\pm 0.6}$	$80.5^{\pm 0.2}$	$78.2^{\pm 0.3}$
FiLM	44.9	46.7	76.5	75.7
MAC	46.3	49.2	81.3	77.4
Vector NMN	48.3	53.5	83.2	77.1
NS-VQA	51.5	52.9	82.5	79.6
LCGN	46.8	48.0	77.0	74.8
ViLBERT	42.4	44.3	69.3	68.7
Visual BERT	41.7	43.2	69.8	63.2
Ours (with BiSAtt)	$61.1^{\pm 0.3}$	$59.7^{\pm 0.2}$	$87.2^{\pm 0.3}$	$83.5^{\pm 0.2}$
Ours (with CoSAtt)	$\mathbf{62.3^{\pm 0.1}}$	$\mathbf{63.3^{\pm 0.1}}$	$\mathbf{89.1^{\pm 0.2}}$	$\mathbf{84.3^{\pm 0.3}}$

Table 5.10: Performance of our language-guided NMN models and state-of-the-art models on S-Ref Dev (S-D), S-Ref Test (S-T), F-Ref Dev (F-D) and F-Ref Test (F-T).

Model	CLS	S-T	F-T
Ours	**88.0**	**63.3**	**84.3**
C1 Ours-L+G	62.1	51.6	77.8
C2 Ours-L-G	61.5	52.1	75.2
C3 Ours+L+(G w/o I)	80.2	57.7	79.9
C4 Ours+L+(G w/o q)	78.8	54.1	76.2
C5 Ours+L+(G w/o m_{arg})	82.1	61.7	80.9

Table 5.11: **Ablations.** Performance of our model with and without LG-Conv layer (L) and CoSAtt encoder (G) on CLOSURE (CLS), S-Ref Test, and F-Ref Test.

performance on CLEVR test split. This demonstrates the effectiveness of the proposed language guided convolutions in capturing visiolinguistic relations and contextual dependencies from the longer CLEVR-like expressions. In particular, we achieve +11.6% in accuracy on CLOSURE test split compared to the best prior model Vector-NMN, indicating that our model generalizes well to unseen compositions. The multi-modal transformer based approaches ViLBERT and VisualBERT performed poorly on both CLOSURE and CLEVR-Ref+, probably due to the mismatched image distribution in pre-training (with conceptual captions [SDG18a]) and fine-tuning. Our model improves the accuracy on CLEVR-Ref+ test splits by 9.8% on S-Ref and 4.7% on F-Ref, compared with the current state-of-the-art method IEP-Ref. Significant gains on S-Test also suggest the superior generalization skills of our model in learning from fewer training samples. Relatively more improvements with CoSAtt encoder compared to BiSAtt encoder shows that early fusion of image and text features facilitate in generating more robust guidance kernel.

To gain better insight into the relative contribution of the design choices we made, we perform experiment with the following five ablated models:

C1: Conv vs. LG-Conv (L). We investigate the contribution of the proposed content adaptive convolution layer in the RCB block by replacing LG-Conv layer with standard convolution. In this setting, we use guidance (G) from CoSAtt encoder for directly scaling up or down the CNN activations in the RCB block.

C2: Conditioning on CoSAtt Guidance (G). In this ablation, we use LG-Conv layers but skip the CoSAtt encoder to verify the importance of module level conditioning on the interaction between image and text features. We instead only pass the module argument (m_{arg}) as guidance to the LG-Conv layer.

C3: CoSAtt w/o. Image (I) We encode guidance using only input query q and the module argument to test the importance of conditioning on image I in the CoSAtt encoder.

C4: CoSAtt w/o. Query (q) We encode guidance using only input image to test the importance of conditioning on input query q in the CoSAtt encoder.

C5: CoSAtt w/o. Module Arg (m_{inp}) In this variant, we keep q and I, but skip m_{arg} in the CoSAtt encoder.

The results are shown in Table 5.11. As we can see, all the above five variants underperform, confirming the importance of our proposed content-adaptive convolutions and guidance kernel.

Results show that module argument in the CoSAtt guidance has less significant effect compared to other components, suggesting that our model is able to infer the semantic context of the module.

5.7 Summary

Neural module networks (NMNs) are widely used in language and vision tasks. In this Chapter, we show that contextualizing these modules dramatically reduces the number of modules required and improve their grounding abilities, achieving a new state-of-the-art results on the CLEVR-Ref+ visual referring expressions task. Our analysis on CLEVR-Ref+, CLOSURE, NLVR2 and a new contrast set CC-Ref+ demonstrate that our proposed method enhances NMNs' ability to exploit visiolinguistic relationships.

5.8 Appendix

In this supplementary material, we begin by providing more details on CLEVR-Ref+ F-Ref / S-Ref splits and the neural modules in IEP-Ref. We then provide the details of our models (e.g., initialization & training, hyper-parameters). Finally, we provide CC-Ref+ dataset annotation details, statistics, random examples, and more analysis.

5.8.1 F-Ref and S-Ref splits in CLEVR-Ref+

Visual referring expression recognition is the task of identifying the object in an image that is referred to by a natural language expression [KOM14, MHT16]. It is a fundamental language-to-vision matching problem and has several downstream applications such as question answering [ZGB16]. CLEVR-Ref+ [LLB19] is a recently proposed dataset for visual referring expression recognition (RefExp) task, which consists of synthetic images and referring expressions. Specifically, it contains the ground-truth functional program representations that describe the intermediate visual reasoning as a chain of logical operations (i.e., neural modules) that need to be executed to find the target referent object (e.g., filter color, compare, filter size, and relate). There are nearly 0.8M referring expressions of which 32% of expressions refer to only a single object (*Single-referent*) and 68% refer to more than one object (*Multi-referent*). In this Chapter, we refer to the full dataset as F-Ref and the single-referent subset as S-Ref. Detailed statistics of the splits are presented in Table 5.12.

		F-Ref	S-Ref
Train	#Expr.	628915	200313 (32% of F-Ref)
	#Images	70000	62016
Dev	#Expr.	69879	22256
	#Images	6500	5200
Test	#Expr.	149741	47731
	#Images	15000	13534

Table 5.12: Statistics of F-Ref and S-Ref.

5.8.2 Neural Modules in Parameterized IEP-Ref

IEP-Ref [LLB19], the current state-of-the-art neural module network (NMN) model for the CLEVR-Ref+ dataset, uses a generic design of neural module architecture adapted from IEP [JHM17b], which was designed for VQA task.[¶] The modules take either two visual inputs (binary modules) or one visual input (unary modules). There are total 60 distinct modules in IEP-Ref. After parameterization (see Figure 5.12b), the distinct number of modules drop to 15 without any drop in the model performance. That is, the number of a distinct set of modules (and the total number of parameters) used in the parameterized model reduces by 75%. Moreover, although the network parameters of each parameterized module slightly increase due to the additional LSTM unit, since each module in IEP-Ref can have multiple instantiations for the same textual input, we have

[¶] We used the IEP-Ref implementation provided at the link `https://github.com/ruotianluo/iep-ref`

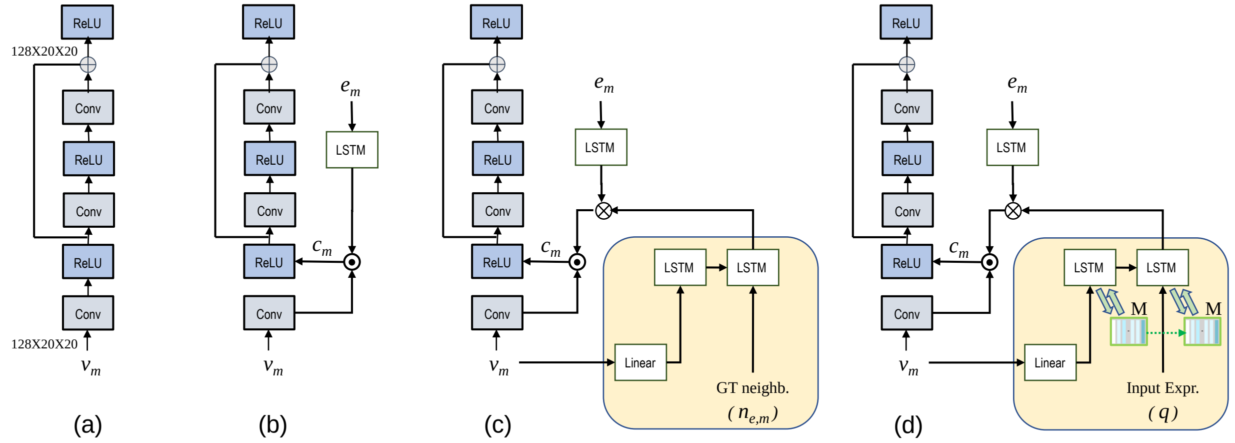


Figure 5.12: (a) Architecture of neural module (m) in IEP-Ref consuming a visual input v_m. $\oplus$ denotes summation. (b) Our proposed module design with parameterized textual input e_m. $\odot$ denotes element-wise multiplication. (c) Contextualized module design using ground-truth annotations for constructing neighborhood context $(n_{e,m})$ of e_m. $\otimes$ denotes concatenation. (d) Contextualized module design using our proposed memory (M) based architecture that learns to select the most relevant neighborhood context directly from the input expression q.

fewer parameters than IEF-Ref in total. Table 5.14 presents the list of all the 15 modules in our parameterized NMN model. We compare the parameters per module of all baseline NMN models and our proposed models in Table 5.13.

Note that our proposed model has 3 times fewer parameters than the baseline model IEF-Ref in total. More concretely, the baseline IEP-Ref model contains 60 modules and each module consists of 0.44M parameters. That is, total number of parameters in IEP-Ref are 60*0.44M = 26.4M. Similarly, the FiLM baseline, which also does contextualization of inputs, has 60*0.59M = 35.4M parameters. On the other hand, our proposed memory based contextualization of NMN model contains only a maximum of 15 modules and each module has 0.58M parameters. Therefore total number of parameters in our model are 15*0.58M = 8.7M. This is 3 times smaller than IEP-Ref and 4 times smaller than FiLM.

Model	#Parameters (per module)
IEP-Ref	442,752
Param. IEP-Ref (P-Ref)	574,336
FiLM	590,720
P-Ref+LSTM+Attn	574,464
P-Ref+CL	574,336
P-Ref+LSTM+Mem	589,597

Table 5.13: Count of parameters for each neural module in the baselines and our proposed NMN models.

	Modules
Unary	Filter_Shape, Filter_Color, Filter_Material, Filter_Visible, Filter_Size, Filter_Ordinal, Unique, Relate, Same_Size, Same_Shape, Same_Color, Same_Material, Scene
Binary	Intersect, Union

Table 5.14: Modules in Parameterized IEP-Ref

5.8.3 Model and other Experiment Details

Our proposed model (LSTM+Mem): The memory matrix consists of 128 rows and 80 columns. The controller is a single layer LSTM network. We use GloVe to obtain the word embedding (dimension = 300) of each word in the textual input. When training, we first train our program generator (PG) and use it as a fixed module for training the execution engine (EE). We use 18K ground-truth programs to train the program generator (PG). We train PG and EE using Adam [KB15] with learning rates 0.0005 and 0.0001, respectively. Note that PG is trained for a maximum of 32,000 iterations, while EE is trained for a maximum of 450,000 iterations. We employ early

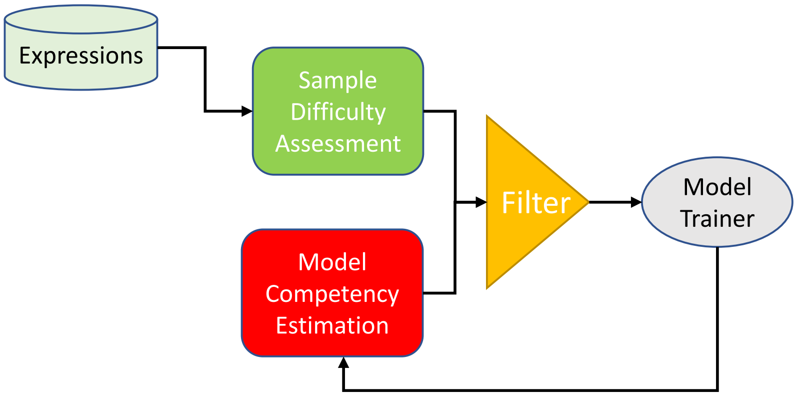


Figure 5.13: Overview of our curriculum learning baseline.

stopping based on validation set accuracy. We do not find any significant improvements with the joint optimization of PG and EE. We train on one RTX 2080ti GPU with a batch size of 8.

Curriculum Learning Baseline: Prior literature shows that curriculum learning (CL) may greatly facilitate the learning of complex tasks for neural architectures [PSN19]. Therefore, we employ a curriculum training (CL) regime as an additional baseline to train the P-Ref model in order to improve its performance without contextualization. An overview of the CL model is shown in Figure 5.13. To estimate the difficulty of the expressions, we define a scoring function inspired by what we, as humans, intuitively may consider difficult when grounding the expressions:

- Longer expressions are difficult to ground.
- Expressions with a large number of spatial relationships such as “left”, “front”, “right”, “behind” are more likely to have difficult linguistic structures.
- Expressions requiring a large number of neural modules are difficult to ground.
- Expressions involving comparison modules are difficult to ground.

Using the above heuristics, we evaluate the difficulty of all expressions in the training set on a scale of 1 to 10. During the training, we initialize the model competency to 1. All the training expressions with difficulty level less than or equal to the current model competency are used for training the model. We use a validation set of expressions for each of these difficulty levels. As the model's performance on the validation set starts to saturate, we increment the competency level of the model. We stop training immediately after the model's competency reaches above 10. We use GloVe to obtain the word embedding (dimension = 300) of each word in the textual input. When training, we first train our program generator (PG) and use it as a fixed module for training the execution engine (EE). We use 18K ground-truth programs to train the program generator (PG). We train PG and EE using Adam [KB15] with learning rates 0.0005 and 0.0001, respectively. PG is trained for a maximum of 32,000 iterations, and EE is trained for a maximum of 450,000 iterations. We employ early stopping based on validation set accuracy. We do not observe any significant improvements with the joint optimization of PG and EE. All of our CL experiments were conducted on one RTX 2080ti GPU with a batch size of 8.

5.8.4 CC-Ref+ Annotation, Statistics, and Visualization

Following gardner2020evaluating, we construct a contrast set for CLEVR-Ref+ dataset to identify systematic gaps (e.g., annotation artifacts) in the test split, and we call it CC-Ref+. Contrast sets help in exposing model brittleness by probing a model's decision boundary local to examples in the test set. We follow a three stage approach to collect our contrast set:

Stage 1: First, we randomly sample 100 single-referent expressions from the test split containing only a single spatial relation (e.g. *The first one of the tiny rubber thing from **left***). We then sample another 100 expressions containing two spatial relations (e.g. *The first one of the thing from **left** that is **behind** the big yellow matte object*). Similarly we sample a third subset of 200 expressions containing 3 or more relations. Finally, we sample 100 expressions containing at least one compare relations (e.g. *Any other tiny object as the **same color** as the big yellow metallic cube*). This constitutes a total of 500 expressions.

Stage 2: We then manually perturb the semantics of various parts of these 500 referring expressions such that the ground-truth referent object changes. For example, we modify the expression *first one*

Referring Expressions	500
Unique Images	492
Vocabulary	86
Expressions with #Relations = 1	100
Expressions with #Relations = 2	100
Expressions with #Relations ≥ 3	200
Expressions with *Compare* Relation	100
Avg. Length of Expression	20.2

Table 5.15: CC-Ref+ Statistics

of the tiny rubber thing from left to *first one of the tiny metallic thing from right*. We show random selection of CC-Ref+ examples in Table 5.16.

Stage 3: Finally, we verify and validate the correctness of the new ground-truth annotations using two human annotators. The annotations that are not consistent among the two human annotators are removed and we re-iterate the above three steps until we collect a validated set of 500 contrast samples[‖]. In Table 6.6, we summarize the size and complexity of our CC-Ref+ split.

Detailed Analysis of Models on CC-Ref+: In the Chapter, we compared the performance of baseline models and our proposed method on CC-Ref+ in terms of number of relations (e.g. *in the front*, *to the left*, *of same shape as*) present in the expressions. In this section, we present more analysis in terms of object attributes. In CLEVR-Ref+, there are six types of object attributes namely, color, size, shape, material, ordinality, and visibility. We analyze the model's performance when one of these attributes are perturbed in the contrast sets. Additionally, we also compare the performance on contrast examples that involve logical AND/OR modifications. An example of

[‖] [GAB20] shows that a few hundreds of contrast samples will be sufficient to draw substantiated conclusions about model behavior.

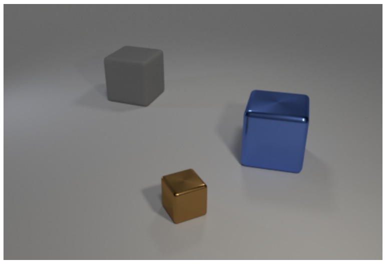
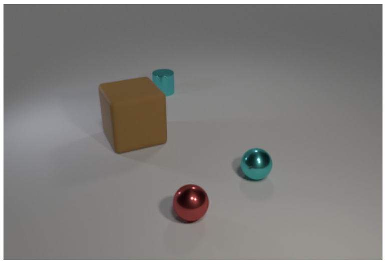
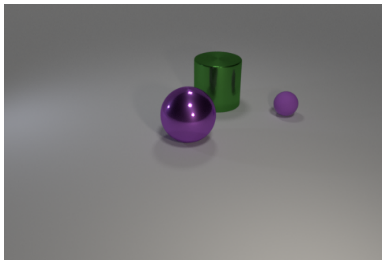
	**Original:** The big objects that are the first one of the block(s) from right or metallic object(s) **CC-Ref+:** The big objects that are the first one of the block(s) from left and rubber object(s)
	**Original:** The brown things that are big object(s) or the second one of the small metal thing(s) from left **CC-Ref+:** The cyan things that are big object(s) or the first one of the small metal thing(s) from left
	**Original:** The small objects that are the third one of the object(s) from left or purple shiny ball(s) **CC-Ref+:** The large objects that are the third one of the object(s) from left or purple shiny ball(s)

Table 5.16: Random examples from CC-Ref+ and their original annotations in CLEVR-Ref+

contrast sample in CC-Ref+ involving logical AND/OR perturbation is as follows:

Original: The objects that are either the first one of the small metal object(s) from right *or* the first one of the metallic cube(s) from left.

CC-Ref+: The objects that are first one of the small rubber object(s) from right *and* the first one of the metallic object from front.

Figure 5.14 shows the performance of baseline IEP-Ref model on original test split and CC-Ref+ samples using the above attributes. Similarly, Figure 5.15, Figure 5.16, and Figure 5.17 shows the performance of models P-Ref+LSTM+Attn, P-Ref+CL, and P-Ref+LSTM+Mem respectively. We found that all the four models are robust to perturbations in color indicating that this is a relatively easier concept to ground in the images. In contrast to the findings in [LLB19], we see a significant drop by up to 15% in the performance of baseline models on all the other attributes such as shape and visibility. P-Ref+CL also experience significant drops in accuracy on CC-Ref+. However it is found to be relatively more robust to the perturbations compared to the other baselines indicating that curriculum learning helps in adapting to contrast sets. Our proposed approach P-Ref+LSTM+Mem shows relatively low drop in the logical, material and ordinal perturbations, insignificant drops ($<$ 3%) in color, visible perturbations and a slight improvement (+2%) in shape perturbations. This clearly suggests that our approach generalizes well and is robust to perturbations in the input. The performance gap of P-Ref+LSTM+Mem in logical, ordinal and material perturbations show that these are relatively difficult concepts for the model to learn. We hope that CC-Ref+ dataset will foster more research in this area.

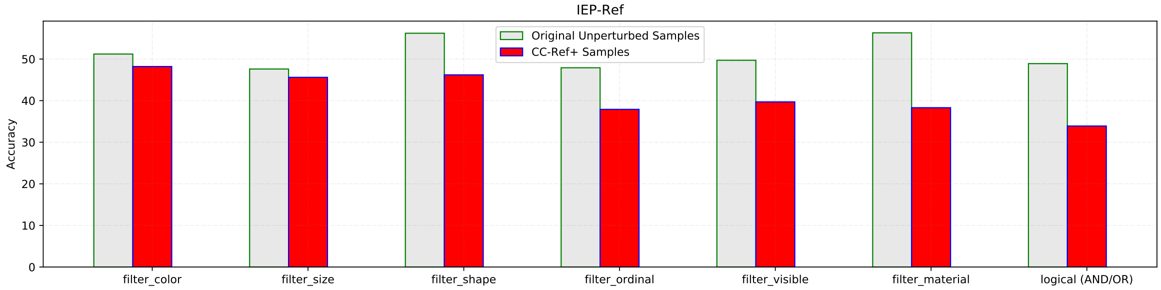


Figure 5.14: Performance of baseline IEP-Ref model on original test split and CC-Ref+ samples

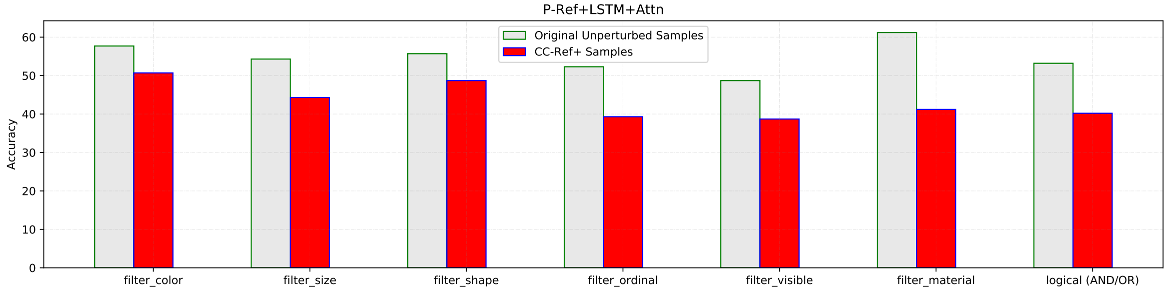


Figure 5.15: Performance of baseline P-Ref+LSTM+Attn model on original test split and CC-Ref+ samples

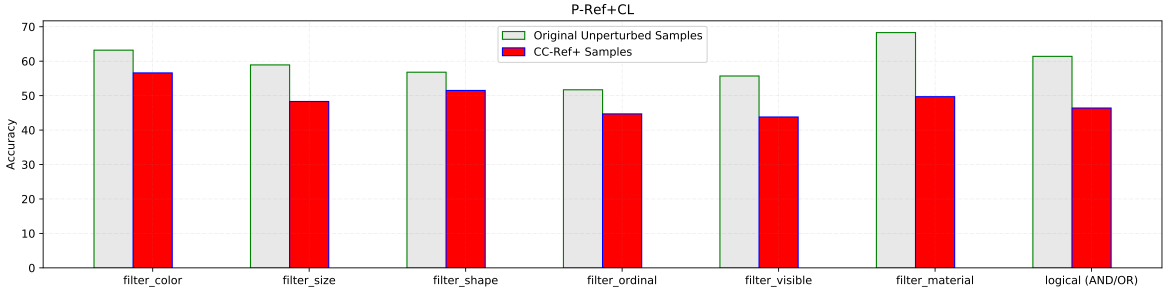


Figure 5.16: Performance of baseline P-Ref+CL model on original test split and CC-Ref+ samples.

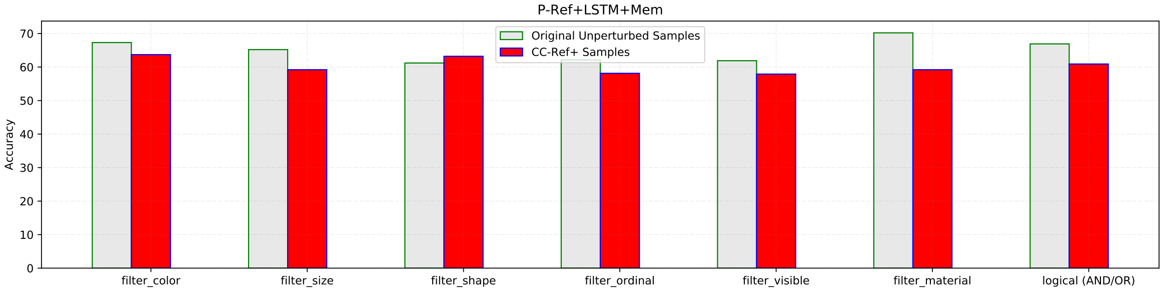


Figure 5.17: Performance of our contextual NMN model (P-Ref+LSTM+Attn) on original test split and CC-Ref+ samples

CHAPTER 6

Model Generalization to Distribution Shifts

In this Chapter, we propose a semi-automatic framework for generating out-of-distribution data to explicitly understand the model biases and help improve the robustness and fairness of existing models [ACG21]. We consider visual question answering (VQA) task to demonstrate the effectiveness of our generation framework.

6.1 Motivation and Objective

One challenge in evaluating visual question answering (VQA) models in the cross-dataset adaptation setting is that the distribution shifts are multi-modal, making it difficult to identify if it is the shifts in visual or language features that play a key role. In this paper, we propose a semi-automatic framework for generating disentangled shifts by introducing a controllable visual question-answer generation (VQAG) module that is capable of generating highly-relevant and diverse question-answer pairs with the desired dataset style. We use it to create CrossVQA, a collection of test splits for assessing VQA generalization based on the VQA2, VizWiz, and Open Images datasets. We provide an analysis of our generated datasets and demonstrate its utility by using them to evaluate several state-of-the-art VQA systems. One important finding is that the visual shifts in cross-dataset VQA matter more than the language shifts. More broadly, we present a scalable framework for systematically evaluating the machine with little human intervention.

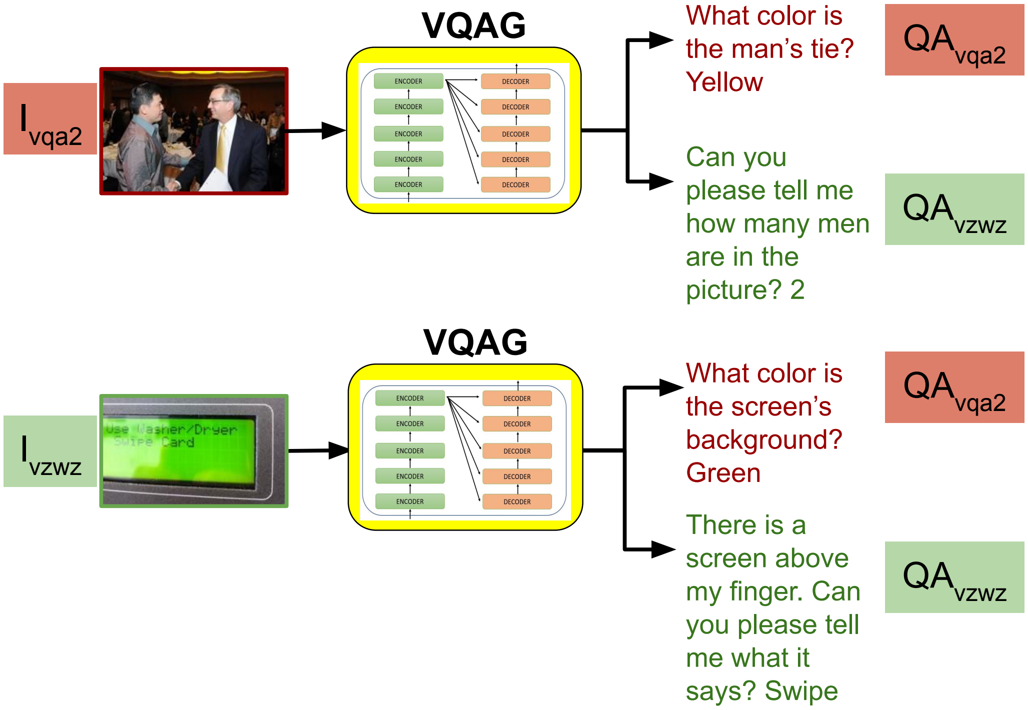

VQAG
I_{vqa2}
What color is the man's tie? Yellow
QA_{vqa2}
Can you please tell me how many men are in the picture? 2
QA_{vzwz}
VQAG
I_{vzwz}
What color is the screen's background? Green
QA_{vqa2}
There is a screen above my finger. Can you please tell me what it says? Swipe
QA_{vzwz}

Figure 6.1: Existing works on VQA domain adaptation between source and target datasets (e.g. VQA2.0 and VizWiz) can only compare the model's performance on the entangled test splits $\langle I_{vqa2}, QA_{vqa2} \rangle$ and $\langle I_{vzwz}, QA_{vzwz} \rangle$. In this work, we propose a VQAG module to generate novel and scalable VQA test sets, called **CrossVQA**, consisting of additional test sets $\langle I_{vqa2}, QA_{vzwz} \rangle$ and $\langle I_{vzwz}, QA_{vqa2} \rangle$ where visual and language features are disentangled.

6.2 Introduction

Multiple datasets have been proposed to measure the progress on visual question answering (VQA) [AAL15, ZGB16, GKS17, GLS18, HM19b, YGL16, TML14, QWL15, LYS16]. However, these datasets often possess biases introduced in the data collection process and by the human annotators. It has been shown that existing VQA models leverage these spurious biases and take shortcuts [GKS17, ABP18, CHS18a, AGA20b]. As a result, the performance of those models on a specific VQA dataset can only serve as a rough proxy for the true learning of the VQA *task* [BSB20].

Test sets	QA_{vqa2}	QA_{vzwz}
I_{vqa2}	✓	✗
I_{vzwz}	✗	✓
I_{oid}	✗	✗

Table 6.1: Existing VQA test sets;

Test sets	QA_{vqa2}	QA_{vzwz}
I_{vqa2}	✓	✓
I_{vzwz}	✓	✓
I_{oid}	✓	✓

Table 6.2: CrossVQA (disentangled) test sets generated by our VQAG model.

One common remedy to this is to go beyond in-domain evaluation, in which the test set exhibits some form of "distribution shifts" from the training set [ABP18, CHS18b]. The key idea is that a generalizable VQA model should be able to extrapolate, for example, from one dataset to another. One challenge that is quite unique to VQA in this setting is that the distribution shift is *multi-modal*. When one dataset unsatisfactorily transfers to another, it is difficult to identify how much of this is due to vision or language distribution mismatches. To complicate things even more, the frequency of objects occurring in natural images follows a long-tail distribution [STT11, ZAR14, ZVF16]. Lack of sufficient instances of minority classes in the test sets further complicates the estimation of generalization capabilities from one dataset to another.

A possible solution to address this issue is to use an iterative, human-in-the-loop approach for dataset collection where human annotators carefully devise new test samples by incorporating visual and language distribution shifts [NWD19, BRW20, GAB20]. However, this approach is not scalable and training the human annotators, be they seasoned AI experts or non-experts, would incur huge annotation time and cost.

In this Chapter, we propose to make the process of creating distribution shifts more systematic and automatic. Inspired by recent work on dynamic benchmarks that co-evolve with strong models [ZHB19], we propose to bring in visual question-answer generation (VQAG) module in the evaluation process. More specifically, we first build a strong, controllable VQAG engine that is capable of creating particular dataset-style question-answer pairs. Then, we use it to generate novel $\langle image, question, answer \rangle$ test splits, while controlling distribution shifts in vision and language features. This is summarized in Table 6.1 and Table 6.2 and exemplified with the VQA2 and VizWiz datasets in Figure 6.1. Collectively, we refer to the resulting VQA test sets as CrossVQA.

There are at least two advantages in using a VQAG model to construct our CrossVQA test sets: (1) We can evaluate the adaptation skills of VQA models on non-VQA datasets such as Open Images (OID) [KRA18], which contains various image annotations but no question/answer pairs, i.e. $\langle I_{oid}, Q_{vqa2} \rangle$ and $\langle I_{oid}, Q_{vzwz} \rangle$ (see Table 6.1); (2) Collecting human-annotated test sets is resource-intensive and scales poorly, while the VQAG approach can be massively scaled and applied in a never-ending learning scenario for generating dynamic benchmarks [NWD19].

We conduct extensive experiments to evaluate the utility of our proposed framework. First, we validate that our VQAG module is capable of generating relevant questions and correct answers with the desired distribution shifts, which we achieve through a combination of transformer-based architectures, vision-and-language pre-training, and multiple types of control signals. We also find that, when evaluated against state-of-the-art generative models for visual question generation, our VQAG substantially outperforms them in terms of accuracy, diversity, and novelty.

Additionally, we perform analysis and human evaluation of our CrossVQA test sets that are built on VQA2, VizWiz, and Open Images datasets. We show that they are effective at finding and quantifying weaknesses of cross-dataset generalization abilities in the state-of-the-art VQA models. For instance, our experimental results show that VQA models drop up to 40% in absolute accuracy if there is a mismatch in image distribution. On the other hand, VQA models are found to be relatively less sensitive to a mismatch in language distribution.

Finally, inspired by the success of contrastive learning and multi-task learning techniques in improving generalization and robustness of multi-modal tasks [AGA20b], we investigate whether these techniques improve the performance of VQA models on our CrossVQA test sets. Interestingly, we find that contrastive losses and multi-task regularization do not lead to significant generalization gains on CrossVQA.

In summary, our key contributions in this Chapter are three-fold. First, we introduce the CrossVQA benchmark for systematically assessing the generalization skills of VQA models, and provide analysis and experiments to support its utility. Second, we describe a scalable data collection and benchmarking framework for semi-automatically constructing the proposed benchmarks using a strong and controllable visual question-answer generation (VQAG) module. Finally, we empirically demonstrate the superiority of our VQAG module by achieving new state-of-the-art results in visual question generation.

6.3 Related Work

Cross-Dataset Distribution Shifts. There is a large body of work analyzing the generalization skills of neural networks from a labeled source domain to a target domain where there is no or limited labeled data [GL15, GSS12, GX12, THD15, AWZ20]. However, these works focus either on language modeling or visual recognition tasks. Here, we investigate adaptation skills using the multi-modal VQA task, for which distribution mismatches can occur in both language and visual features.

There are a few works that study systematic compositional skills in multi-modal tasks. For example, Lampert et al. [LNH09] study the use of attributes in transferring information between object classes. Jabri et al. [JJV16] explore several variants of the VQA task and show that VQA models struggle with transferring knowledge across datasets. Agrawal et al. [ABP18] study the extent to which a model is visually grounded, by evaluating its ability to generalize to a different answer distribution for each question type. Chao et al. [CHS18b] investigate the issue of cross-dataset generalization, using a specific setting where the source domain contains a large amount of training data and the target domain contains insufficient data to train a VQA system from scratch. Unlike these works, our work performs a more fine-grained analysis by disentangling the distribution mismatches in language and vision, achieved by generating out-of-distribution shifts using a learned VQAG module.

Visual Question Generation (VQG). The goal of VQG is to generate natural questions for an image. This task has drawn much attention due to its ability to test a model's understanding of natural language in the context of visual grounding and its application in downstream tasks such as image retrieval and question answering [AAL15, ZGB16, Aku15, PRA15].

While the task of generating question automatically is well studied in the language domain, it has been under-explored for image-related natural questions [MMD16]. Prior works explored VQG using autoencoder-based architectures [JZS17, YLL18, ALC19, KBF19]. Jain et al. [JZS17] employ a variational autoencoder paradigm where they first learn to embed a given question and image into a low dimensional latent space. The latent codes are subsequently mapped to a high-dimensional representation using RNNs during inference to generate the question. Krishna et al. [KBF19] model question generation as a process that maximizes mutual information between the image and the expected answer's category. They incorporate fine-grained answer type as the guidance to generate goal-driven questions. Xu et al. [XWY20] propose an answer-centric approach where they model the complex relationship between an answer and its relevant image regions. Unlike these works, our approach uses a simple encoder-decoder framework, but we enhance it using a transformer-based architecture, vision-and-language pre-training, and various control signals, which together lead to a stronger VQG model. Furthermore, our work not only improves the VQG performance, but also takes a step further by exploring *using* VQG in the context of VQA evaluation.

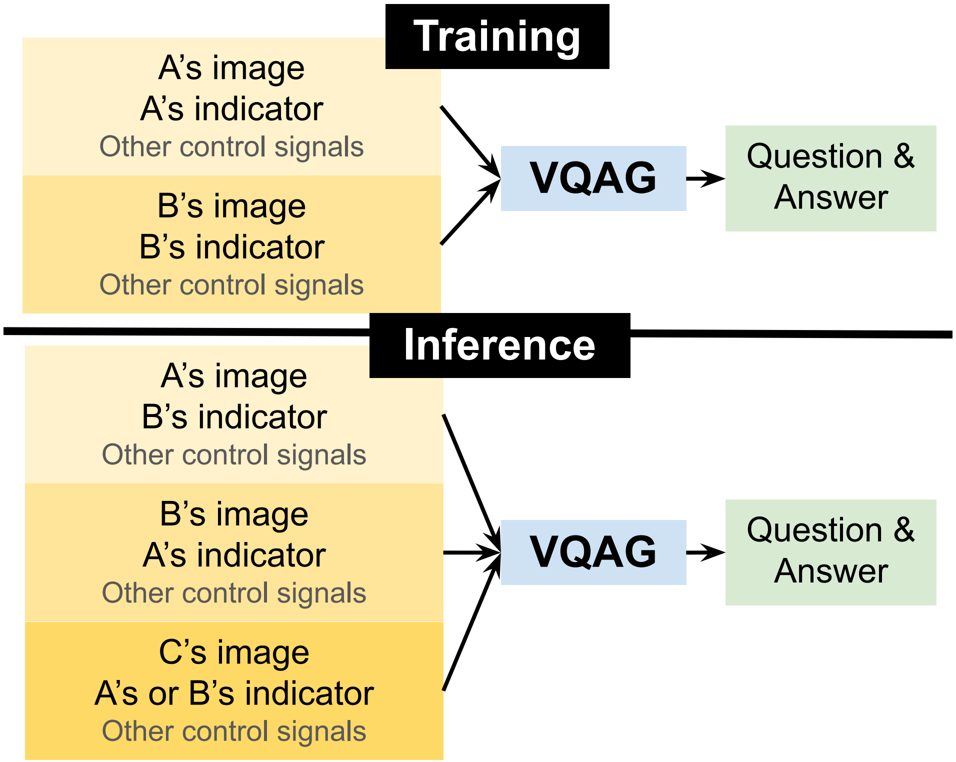


Figure 6.2: **Overview of CrossVQA**. We train a controllable visual question-answer generation (VQAG) engine and use the dataset indicators and control signals to generate the desired cross-dataset shifts.

6.4 Approach

6.4.1 Overview

Figure 6.2 overviews our approach to systematically generating cross-dataset distribution shifts. During training, we train a visual question-and-answer generation (VQAG) engine using multiple sources of VQA data (denoted by A and B). This VQAG module uses a dataset indicator to learn and generate question-answer pairs of a particular dataset's style.

During inference, we apply the trained VQAG model to multiple image sources (denoted by A, B, and C), while varying the dataset indicator. For example, we turn on the dataset B indicator for the images of A, which generates B-style questions/answers for the images in A. Furthermore, VQAG can also be applied to images from a different dataset C, for which no VQA annotations are available, yet we can still control the style of annotations generated. In the post-processing step, the resulting VQA datasets are validated by human annotators.

We first provide more details on our VQAG engine (Sec. 6.4.2) and then describe how it is used to generate CrossVQA benchmarks (Sec. 6.4.3).

6.4.2 Visual Question-Answer Generation

We start from a transformer-based encoder-decoder model that learns to generate question-answer pairs from images. We then enhance this model in two ways. First, we perform image-text pre-training using a recently introduced Conceptual 12M (CC12M) dataset [CSD21]. Second, we experiment with multiple control signals. As we will show in our experimental results, these signals help improve the accuracy and the diversity of the generated outputs when applied to diverse sources of images.

Base VQAG Model and Input-Output Format. We adopt a transformer-based encoder-decoder framework [VSP17] for image-to-text generation as our base model, following recent work on large-scale image captioning [SDG18b, CPS19]. In particular, we represent each input image as a sequence of feature vectors, and the model learns to produce relevant questions and their corresponding correct answers.

Each input image is represented by multiple types of visual features [CPS19], which we briefly describe here (see Appendix for more details):
(i) a global feature vector extracted by Graph-RISE [JLL19], a ResNet-101 [HZR16] trained for image classification at ultrafine granularity levels;
(ii) 16 regional feature vectors, obtained from Graph-RISE featurization of top-16 proposals of a Faster RCNN [RHG16] object detector trained on Visual Genome [KZG17];
(iii) top semantic object label vectors, where labels (e.g. “river”, “man”, “football”) are produced

Table 6.3: Dataset-agnostic control signals

Notation	Description
P	Question prefix
C	Answer category
A	Most common answer
Ã	All answers

by the Google's Vision API[*].

Our target is a question-answer pair in the format $q \langle sep \rangle a$, where q is the question tokens, a is the answer tokens, and $\langle sep \rangle$ is the chosen delimiter. Furthermore, since a is not limited to a single answer [BLG19], a is represented as $a_1\langle dsep\rangle a_2\langle dsep\rangle \dots \langle dsep\rangle a_K$, where $a_1, a_2, \dots, a_k$ are possible answers for q. We use beam search to generate the target question and answer(s) during the decoding stage.

Next we incorporate two enhancements into this base model to (a) maximize the relevance between image, question and expected answer in the generated test sets; (b) improve generalization capability of the model to out-of-domain images; and (c) increase the diversity and novelty of the questions.

Enhancement 1: Image-To-Text Pre-Training. We pre-train our base VQAG model on Conceptual 12M [CSD21], a large-scale dataset specifically designed for vision-and-language pre-training. It consists of 12.4 million image–Alt-text pairs harvested from the Web. We use the standard image captioning objective for pre-training [CSD21]. Despite this task mismatch (i.e., image captioning vs. visual question/answer generation), we observe the utility of pre-training in addressing the long-tail distribution of objects (see Sec. 6.5.2)

Enhancement 2: Dataset-Agnostic Control Signals. In addition to the image features, we also

[*] https://cloud.google.com/vision/docs/labels

Table 6.4: Examples of Dataset-agnostic control signals.

Examples
Question 1: Is the screen's background blue? **P** : *Is the*, **C** : *Color*, $\tilde{\mathbf{A}}$: *yes <dsep> true <dsep> blue screen <dsep> yes,* **A** *: yes*
Question 2: How many men are in the picture? **P** : *How many*, **C** : *Counting*, $\tilde{\mathbf{A}}$: *2 <dsep> 2 <dsep> 3 <dsep> 5,* **A***: 2*

condition our model on up to three control knobs more directly related to visual question generation and answering. In particular, we explore three main types of dataset-agnostic control signals, summarized in Table 6.3: the expected first two words of the question (i.e. question prefix), the expected answer category, and the expected answer(s). See Appendix for further discussion.

To condition the VQAG model on these control signals, the embeddings for the control signals are fed to the encoder together with the image embeddings. The visual and language features from the image embeddings and the control signals are allowed to attend to all other features through the self-attention mechanism.

Dataset indicator as additional control signal. As the main focus of this paper is cross-dataset shifts, we consider the dataset indicator control signal as an additional input. This signal helps inform the model of the desired domain or style of visual questions. Similar to dataset-agnostic control signals above, the one-hot embedding for the dataset indicator is concatenated to the image and other control signal embeddings and fed to the encoder.

6.4.3 Generating CrossVQA Benchmarks

We now describe how to use the enhanced VQAG model together with the dataset indicator described in previous section for generating CrossVQA benchmarks.

Datasets. We consider two VQA datasets: VQA2 [GKS17] and VizWiz [GLS18]. The two datasets are drastically different visually and textually. VQA2 is built on top of high-quality COCO images [LMB14b] with visual questions intended to fool "smart robot" but not humans. VizWiz, on the other hand, is collected in-the-wild from the visually-impaired users, often with lower image quality and more conversational and simpler questions intended to be useful if answered correctly.

Additionally, we consider the images from Open Images (OID) [KRA18], which is known to have more diverse objects than COCO [ADW19].

6.4.3.1 Training

We mix the training splits of VQA2 and VizWiz and use that for training our VQAG. We experiment with pre-training and different combinations of dataset-agnostic control signals (Sec. 6.5). We leverage ground-truth control signals in the training set whenever available; question prefixes and answers are available for both datasets, while the answer categories are available on a subset of VQA2, as provided by [KBF19].

6.4.3.2 Inference

Creating Disentangled Shifts. By varying the dataset-indicator control knob of our best-performing VQAG models, we generate our desired disentangled shifts. More specifically, denote by $\langle I_A, QA_B \rangle$ a dataset with A-style images and B-style questions. We generate the following four VQA splits: VQA2-style question-answer pairs on a subset of VizWiz validation images $\langle I_{vzwz}, QA_{vqa2} \rangle$, VizWiz-style question-answer pairs on a subset of VQA2 validation images $\langle I_{vqa2}, QA_{vzwz} \rangle$, and additionally both VQA2-style and VizWiz-style pairs on a subset of OID validation images $\langle I_{oid}, QA_{vqa2} \rangle$ and $\langle I_{oid}, QA_{vzwz} \rangle$. In addition, we also generate $\langle I_{vqa2}, QA_{vqa2} \rangle$ and $\langle I_{vzwz}, QA_{vzwz} \rangle$ as a sanity check to verify if our model learns to understand the styles of VQA2 and VizWiz.

Dataset-agnostic control signals. There are no ground-truth control signals for the images during inference. Thus, we train an image tagger with the multi-label sigmoid cross entropy loss to predict top-k most relevant first two words (i.e. question prefix), answer categories, and answers from the input image and the target dataset indicator This is more flexible than the approach used in [KBF19] where all the pre-annotated answer categories are used during inference for all images.

6.4.3.3 Postprocessing

We further clean CrossVQA by using the human annotators to assess question relevance and answer correctness (Sec. 6.5.2).

6.5 Experiments

In this section, we first evaluate the performance of our VQAG model against existing state-of-the-art baselines [KBF19, WYT17, JZS17]. We then demonstrate the importance of conditioning our VQAG model on the proposed control signals by performing several ablation studies. Next, we present CrossVQA examples and several statistics based on the generated data. We finally show that CrossVQA is effective at identifying the limitations of state-of-the-art VQA models, and examine the extent to which existing adaptation techniques help in improving performance of VQA models as measured by CrossVQA.

6.5.1 In-Domain Evaluation of VQAG

We first benchmark the in-domain performance of our VQAG model by training and testing on VQA2 [GKS17] against existing models for visual question generation (VQG). Note that, unlike those models which focus on generating only questions, our model also generates answers; we discard the generated answers when evaluating the generated questions against existing work.

Metrics. We consider two sets of evaluation metrics. The first set of metrics measure **question relevance**. It consists of multiple automatic text similarity metrics widely used for image captioning and VQG: BLEU [PRW02], ROUGE-L [Lin04], METEOR [BL05], SPICE [AFJ16] and CIDEr [VLP15]. The second set of metrics measure the **diversity and novelty** of questions and answers [VCS16, JZS17]: (i) generative strength: the percentage of unique generated questions normalized by the number of unique ground truth questions, (ii) inventiveness: the percentage of unique generated questions that are unseen during training, (iii) oracle CIDEr: the maximum value of the CIDEr over a list of all references. Note that, although not considered by previous work, both generative strength and inventiveness for questions (QS and QI, respectively) can be extended to measure the diversity and novelty of generated answers as well (AS and AI, respectively).

Notation. We use X2Y to denote the model with X as input and Y as output. We use I, Q, A, C to refer to image, question, answer, and answer category, respectively. Furthermore, we use Ã to refer to multiple answers and P to question prefix. See Table 6.3 for examples of our control signals.

Baselines. We compare the performance of our VQAG model against the following baselines: **IA2Q** [WYT17], a non-variational model that takes an image and answer as input and generates a question; **V-IA2Q** [WYT17], a variational-autoencoder based approach that embeds the input image and question to a latent space before generating a question; **IC2Q** and **V-IC2Q**, extensions to the IA2Q and V-IA2Q models, respectively, where the models are conditioned on answer categories [KBF19] instead of ground-truth answers; **MI-IA2Q** [KBF19] and **MI-IC2Q**, also variational models posing the question generation as a process that maximizes mutual information between the image, the expected answer and the answer category.

Results. Results are reported in Table 6.5. Our models (IÃC2QÃ, IÃP2QÃ) significantly outperform all the baselines on standard automatic metrics by large margins, especially improving the BLEU-4, METEOR and CIDEr scores by +29.5%, +23.17% and +0.62, respectively, compared to the current state-of-the-art methods MI-IC2Q and MI-IA2Q. In addition, our best model IÃP2QÃ outperforms the state-of-the-art MI-IC2Q by +7.06% in QS, suggesting that we generate a diverse pool of questions. Moreover, for question inventiveness, a +30.39% QI improvement paired with a high oracle CIDEr score indicates that our model also generates novel and appropriate questions by using new combinations of objects and question patterns. We also find a +20% improvements in AS and AI with the enhancements discussed in Sec. 6.4.2.

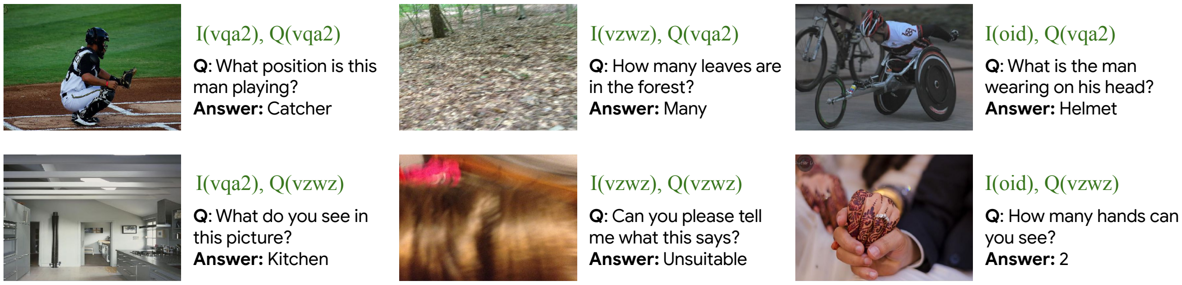

I(vqa2), Q(vqa2)
Q: What position is this man playing?
Answer: Catcher
I(vzwz), Q(vqa2)
Q: How many leaves are in the forest?
Answer: Many
I(oid), Q(vqa2)
Q: What is the man wearing on his head?
Answer: Helmet
I(vqa2), Q(vzwz)
Q: What do you see in this picture?
Answer: Kitchen
I(vzwz), Q(vzwz)
Q: Can you please tell me what this says?
Answer: Unsuitable
I(oid), Q(vzwz)
Q: How many hands can you see?
Answer: 2

Figure 6.3: Qualitative examples of questions and answers in our CrossVQA dataset.

6.5.2 Analysis of Generated Data

Now that we establish the superiority of our VQAG engine to existing approaches, we analyze the outputs of our best model (IÃP2QÃ with pre-training) when used to generate CrossVQA benchmarks (Sec. 6.4.3.2).

Statistics and Examples of CrossVQA. Table 6.6 presents basic statistics of the six CrossVQA test splits generated by our VQAG model. Figure 6.3 provides examples.

Human Evaluation. We first conduct a human study to verify **question relevance and answer correctness** of 3000 samples from the generated splits. More concretely, we present each <image, question, answer> triplet to three crowd workers and ask them to verify if the generated question is relevant to the image. Questions that are annotated as not relevant by at least two workers are discarded. For each of the relevant questions, we also ask the workers to verify if the generated answer is correct, and, if incorrect, ask them to write a correct answer (See Appendix).

As shown in Table 6.7, workers annotate a large portion of the generated questions by our VQAG model as relevant (QR percentages between 77.4% and 97.8%), showcasing the effectiveness of the proposed VQAG model. Answer correctness is found to be relatively lower (AC percentages between 51.6% and 74.8%), a result that indicates that CrossVQA is a challenging new benchmark for visual question answering. We find that the questions belonging to *count*, *time*, *spatial*, *food* and *attribute* categories are relatively more difficult for our model to generate correct answers.

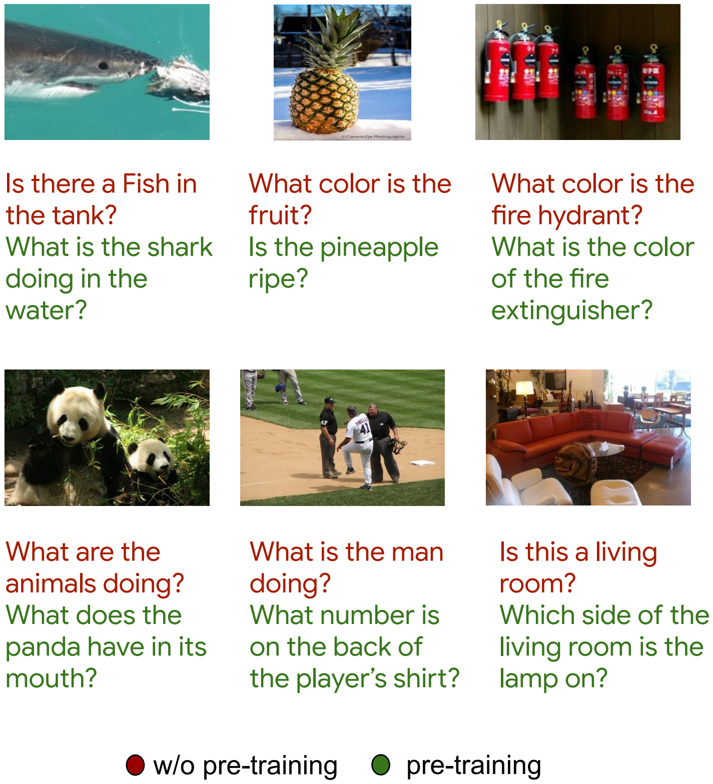

Is there a Fish in the tank?
What is the shark doing in the water?
What color is the fruit?
Is the pineapple ripe?
What color is the fire hydrant?
What is the color of the fire extinguisher?
What are the animals doing?
What does the panda have in its mouth?
What is the man doing?
What number is on the back of the player's shirt?
Is this a living room?
Which side of the living room is the lamp on?
🔴 w/o pre-training 🟢 pre-training

Figure 6.4: Pre-training improves the ability of the VQAG model to generate questions and answers about long-tail concepts (images in the figure are from OID).

Further Analysis. We first assess the controllability ability of our VQAG model in the generation of VQA2-style or VizWiz-style questions. In Table 6.8, we use the Jensen-Shannon (JSD) divergence between the unigrams and bigrams distributions of questions between each data pair to measure their "style" distance. Regardless of the image sources, the generated VQA2-style (VizWiz-style) questions are much more similar to VQA2 (VizWiz) than the original VizWiz (VQA2) questions are.

We then focus on the generated questions/answers on OID and assess the benefits of pre-training and control signals on out-of-domain images. Figure 6.4 shows a qualitative comparison of questions generated without (red) and with pre-training (green). We observe that the pre-trained model generates more accurate and informative questions (e.g., *fire hydrant* vs. *fire extinguisher*,

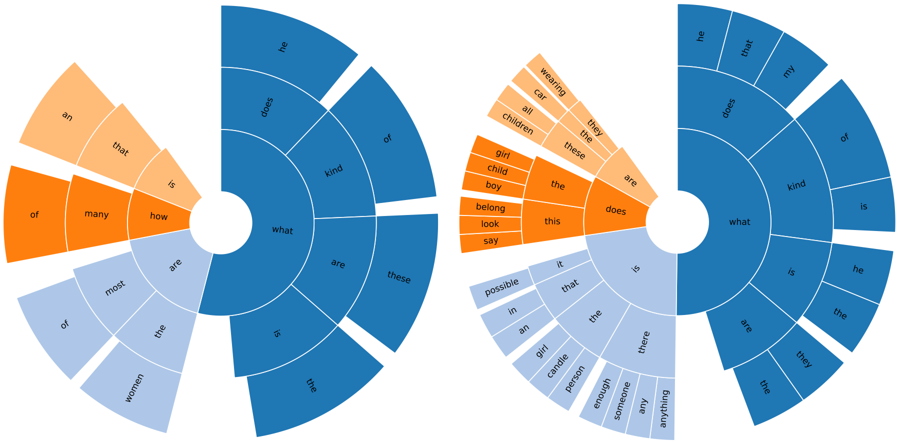


Figure 6.5: Distribution of the first three words for questions generated without (left) and with (right) control signals (on OID).

fish vs. *shark*). In Figure 6.5, the sunburst plots (shown at the top) of the first three words of the questions exhibit much higher diversity with control signals. Further, in Figure 6.6, the distribution of answer categories demonstrate that control signals increase the entropy of answer category distribution, helping the heavy tail ones.

6.5.3 Cross-Dataset VQA Experiments

Performance of Existing VQA Systems on Human-Validated CrossVQA. On the 2100 human-validated CrossVQA relevant questions, we evaluate the VQA adaptation performance of the state-of-the-art VQA models: ViLBERT (VB) [LBP19b], LXMERT [TB19], and VisualBERT [LYY19], all trained on VQA2. In Table 6.9 (top three rows), we find that ViLBERT outperforms other baselines on CrossVQA splits with VQA2 images or VQA-style questions, so we provide a detailed analysis of ViLBERT.

Figure 6.7 compares the CrossVQA performance of (a) ViLBERT trained on the VQA2 dataset, and (b) ViLBERT trained on VQA2 and fine-tuned on VizWiz. We find that both VQA models show accuracy drops on all six splits, compared to the SOTA accuracy 71.0% on VQA2 test set and 54.7% on VizWiz test set (left-most column). This indicates that the questions in CrossVQA are

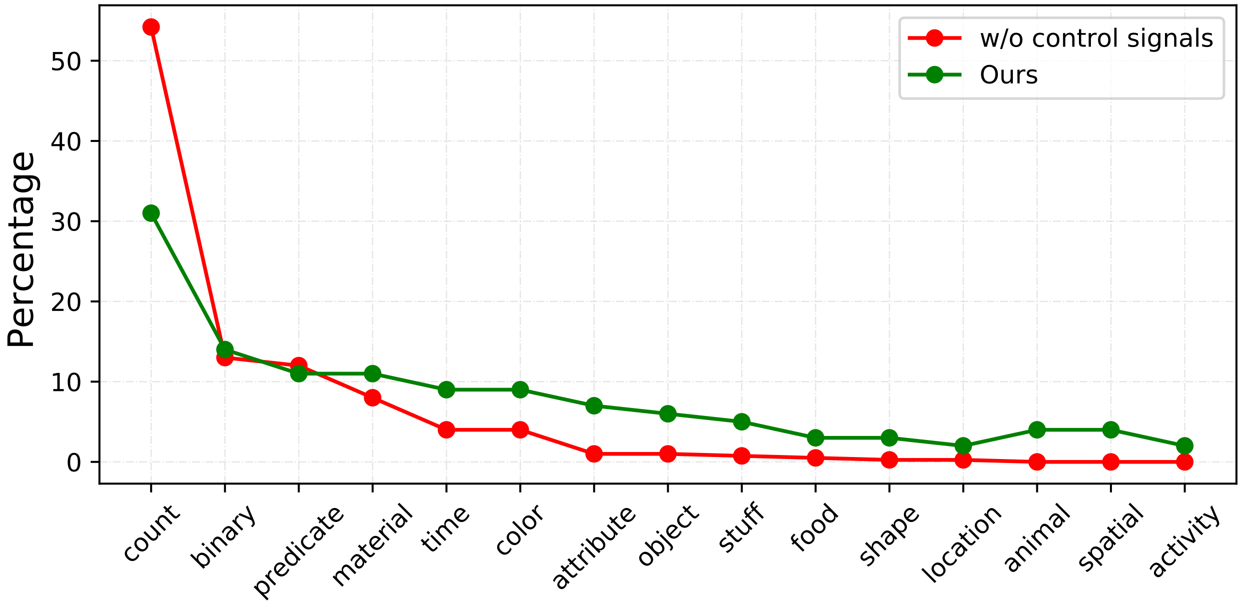


Figure 6.6: Distribution of answer categories generated without (red) and with (green) control signals (on OID).

harder for SOTA models to get right. Moreover, the model trained on VQA2 drops by up to 40% on VizWiz and OID input images, a rather unexpected (and never-before quantified) result. Similarly, the model trained on VizWiz underperforms on splits with VQA and OID images by similarly large margins. This suggests that the VQA models struggle to generalize when there is a mismatch in image distribution. In contrast, the drop in accuracy is relatively low for mismatches in language distribution, indicating that these models are relatively less robust to visual features compared to language features. We believe that the rich object-level features and interactions available in the visual space could be causing the models to overfit to training image distribution and therefore the models struggle to generalize to new image distribution.

Adaptation Techniques with Auxiliary Losses. We also examine if the contrastive and multi-task (MTL) losses [AGA20b] improve the adaptation performance of ViLBERT on CrossVQA in Table 6.9. In contrastive leaning, negative examples that are close to the current example are mined, and used to learn to jointly minimize the loss on the current (positive) example and maximize the loss on the (hard) negative examples. Two versions of contrastive losses are considered: Sum of

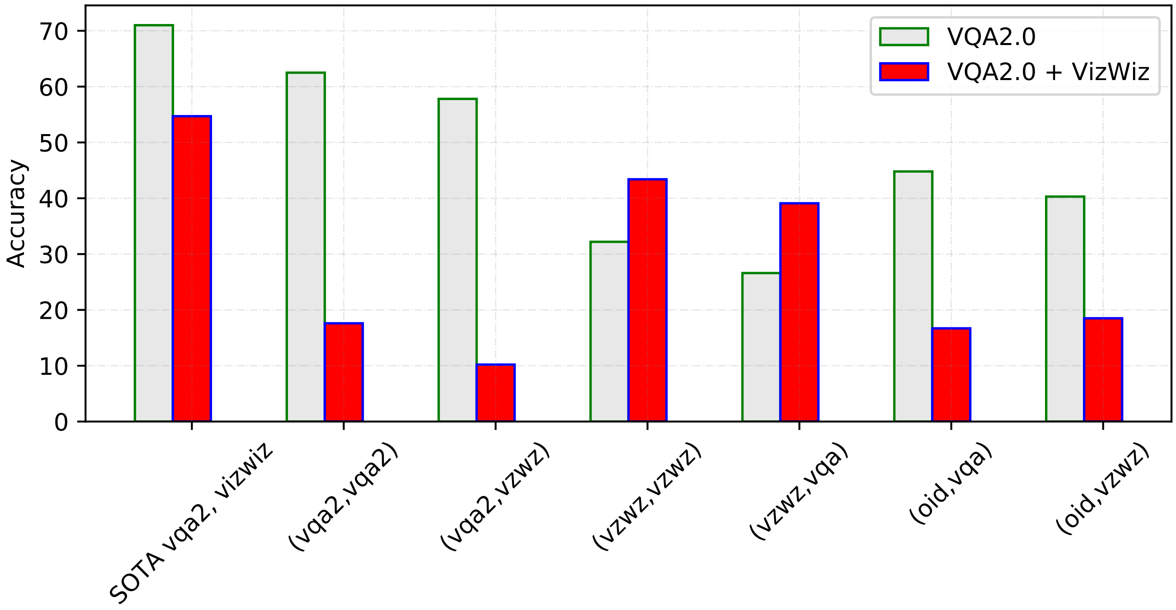


Figure 6.7: The CrossVQA performance of ViLBERT, trained on VQA2 only (VQA2.0) or trained on VQA2 and then fined-tuned on VizWiz (VQA2.0 + VizWiz). Left-most column indicates the reference state-of-the-art performance.

Hinges (Sum-H), taking a sum over all negative samples, and Max of Hinges (Max-H), which only considers the loss on hardest negative sample by applying the max operation. For MTL, the following auxiliary tasks are used: GQA [HM19b], visual common sense reasoning (VCR) [ZBF19], and referring expression recognition with RefCOCOg (RER) [MHT16]. The last five rows of Table 6.9 show the performance of ViLBERT (VB) using these contrastive and MTL losses. Although the losses slightly improve the accuracy on in-domain CrossVQA split $\langle I_{vqa2}, QA_{vqa2}\rangle$, they fail to improve generalization on cross-domain splits $\langle I_{vqa2}, QA_{vzwz}\rangle$, $\langle I_{vzwz}, QA_{vqa2}\rangle$ and $\langle I_{oid}, QA_{vqa2}\rangle$, suggesting that there is ample room for improvement (see Appendix).

6.6 Summary

In this Chapter, we present a step toward scalable and systematic evaluation of VQA systems. Key to our approach is an accurate and controllable VQAG module that is capable of generating disentangled distribution shifts. We generate CrossVQA benchmarks, a collection of test splits based on VQA2, VizWiz, and Open Images datasets. We validate their utility by showing that existing VQA models struggle to perform well in this evaluation scenario and identifying the image distribution mismatch as the main factor.

6.7 Appendix

6.7.1 Implementation Details

The models are optimized with Adam [KB15] with an initial learning rate of 0.000032. We use a linear decay learning rate schedule with warm up and employ early stopping based on validation set accuracy. If not pre-trained, we train our VQAG model for a maximum of 2M iterations. With pre-trained initialization, we train our VQAG model for a maximum of $500,000$ iterations. Both the encoder and decoder layers of transformer have 6 layers each with 8 heads for multiheaded attention. The vocabulary embedding size is 512, and the hidden embedding size is 1024. We train our models with a global batch size of 4096 over Google Cloud 32-core TPUs[†]. The average training time for pre-training on conceptual captions dataset is 52 hours, and training on VQA2.0 and VizWiz takes up to 21 hours.

We condition our VQAG model using the expected answer categories (Ã) of the output answer as one of the control signals, in order to maximize the relevance between image, question and expected answer in the generated test sets. These answer categories can be objects, attributes, colors, materials, time, etc. Specifically we use 16 categories (similar to [KBF19]), covering more than 80

[†] https://cloud.google.com/tpu/

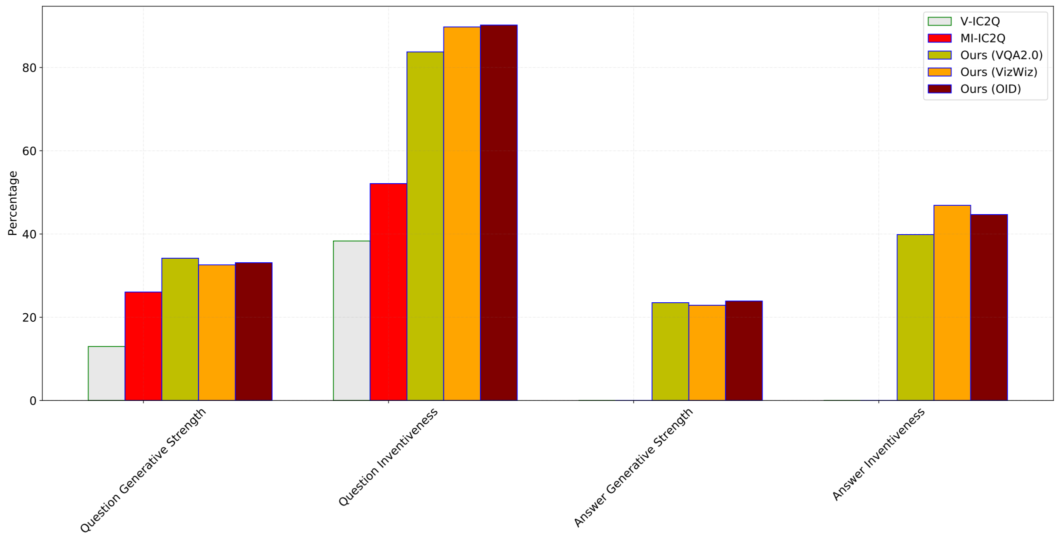


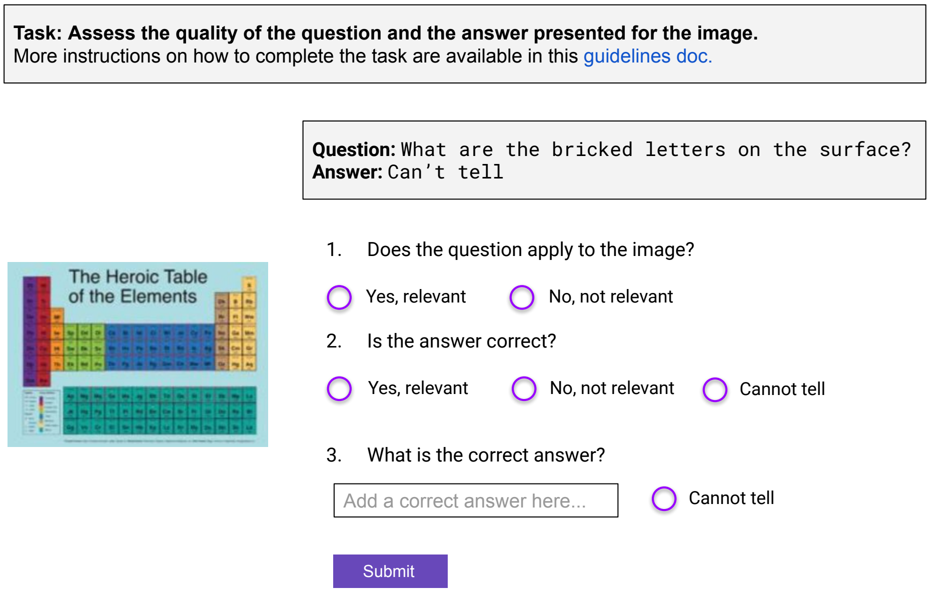
Figure 6.8: Diversity and Novelty of our VQAG model on out-of-domain splits: VizWiz val split and OID val split.

objects, 40 attributes, 17 colors, and 8 materials. Table 6.10 presents the list of all the 16 categories and provides examples of answers for each of the categories.

The decoder generates the question and the answer(s) separated by delimiters, for example, question $\langle sep \rangle$ answer1 $\langle dsep \rangle$ answer2. We use beam search (width = 5, alpha = 0.6) to generate the target question and answer(s) during decoding.

6.7.2 More Results on Diversity and Novelty

In Section 4 of the main paper, we show that control signals improve the diversity and novelty of the generated questions through the metrics question generative strength (QS) and inventiveness (QI), answer generative strength (AS) and inventiveness (AI). To do this, we trained our VQAG model on VQA2.0 train split and evaluated the model performance on the in-domain VQA2.0 val split. In this section, we additionally show the performance of VQAG model on out-of-domain (o.o.d) splits, namely, VizWiz val split and OID val split. Figure 6.8 shows the results. As we can see, there is no significant drop in QS and AS on o.o.d splits, suggesting the superior generalization



Task: Assess the quality of the question and the answer presented for the image.
More instructions on how to complete the task are available in this guidelines doc.

Question: `What are the bricked letters on the surface?`
Answer: `Can't tell`

1. Does the question apply to the image?

○ Yes, relevant ○ No, not relevant

2. Is the answer correct?

○ Yes, relevant ○ No, not relevant ○ Cannot tell

3. What is the correct answer?

Add a correct answer here... ○ Cannot tell

Figure 6.9: Experiment interface for human evaluation to verify question relevance and answer correctness.

skills of our model. Moreover, increase in QI and AI indicates that model is relatively more creative in inventing new questions and answers on o.o.d splits compared to in-domain splits. Table 6.11 presents examples of the invented/unseen questions and answers that are not seen by our VQAG model during training. In the next section, we verify the question relevance and answer correctness of these o.o.d questions.

6.7.3 Additional Human Evaluation Results

We verify question relevance and answer correctness of the samples in CrossVQA splits where the VQAG model is trained on combined train sets of VQA2.0 and VizWiz. In this section, we present additional results on human evaluation of VQAG model that is trained on only VQA2.0 train split. We generate questions and answers for VQA2.0 val split (in-domain) and VizWiz, OID val splits (o.o.d). Figure 6.9 shows the interface used for conducting this study. Questions that are annotated as not relevant by at least two workers are considered as irrelevant. For each of the

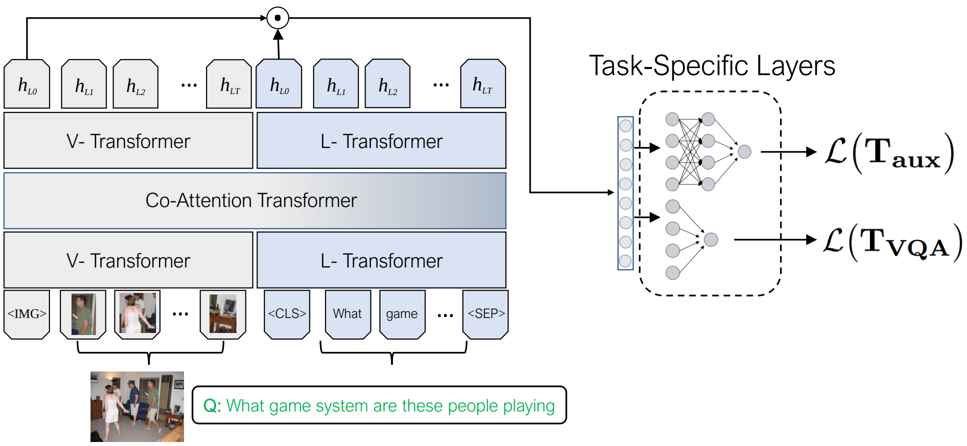


Figure 6.10: Multi-task learning model for VQA with auxiliary tasks such as GQA, REF, and VCR.

relevant questions, we ask the workers to verify if the generated answer is correct, and if incorrect, ask them to write the correct answer. Table 6.12 present human evaluation results. A significant portion of generated questions are annotated as relevant. Moreover, we do not find significant differences in QR and AC metrics across in-domain and o.o.d samples, confirming that the higher percentage of invented questions on o.o.d splits (in Figure 6.11) are indeed relevant and not due to random noise. Furthermore, in Table 6.12, we also show the QA and AC percentages across seen and unseen questions generated by VQAG model. We see higher drop in AC percentage on unseen questions compared to the drop in QR, indicating that unseen questions are relatively harder for the model to generate correct answers.

6.7.4 More Details on our Base Model

Both the encoder and the decoder contain a stack of L layers, with each layer consisting of a multi-head self-attention layer followed by a feedforward layer. For a given token embedding, the self-attention layer produces a weighted representation of all other tokens in the input. This weighted representation is then combined with the input representation of the given token and it is passed to the next layer.

Specifically, each attention head first calculates the queries Q, keys K and values V as follows:

$$Q = XW_Q, K = XW_K, V = XW_V \tag{6.1}$$

where X contains all the input features stacked into a matrix, and W_Q, W_K, and W_V are learned projection matrices.

The output of the attention head is then computed as follows:

$$\text{ATTN}\left(Q, K, V\right) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V \tag{6.2}$$

where d_k, d_v are the dimension of the keys K and values V respectively. Intuitively, with the above attention, the encoder jointly attends to information from different representation subspaces at different positions in the input image.

The point-wise feedforward network (FFN) is applied to each output of the attention layer and it consist of two linear transformations, with a ReLU activation in between,

$$\text{FFN}\left(x\right) = max\left(0, xW_1 + b_1\right) W_2 + b_2 \tag{6.3}$$

where W_1, b_1 and W_2, b_2 are the weights and biases of two fully connected layers.

Embedding Regional Image Features We extract image objects and their features using a Faster RCNN [RHG16] object detector model, trained on Visual Genome [KZG17]. We extract 100 object regions per image. The resulting bounding boxes are considered as *visual tokens*. Similar to the positional encoding in language models [VSP17], for each visual token, the spatial position of bounding box is also encoded. We use a 5-d vector, $p_{spatial}$, to encode the top-left, bottom-right, and the bounding box area relative to the image, i.e., $p_{spatial} = [\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{x_{br}}{H}, \frac{w \cdot h}{W \cdot H}]$.

Embedding Global Image Features Similar to [TS20, CPS19, PUC19], we also use a global image representation using the Graph-RISE model [JLL19], a ResNet-101 model [HZR16] trained for image classification at ultrafine granularity levels. These regional and global image features $f_I = (f_r, f_g)$ are fixed during training.

$$\begin{aligned} f_r &= \text{RCNN}(I; \theta_{RCNN}) \\ f_g &= \text{GraphRISE}(I; \theta_{GraphRISE}) \end{aligned} \tag{6.4}$$

6.7.5 Models for Adaptation Analysis

ViLBERT Training: As discussed in Section 4 of the main paper, we use **ViLBERT** [LBP19a] for our adaptation experiments. ViLBERT uses a pretrain-then-transfer learning approach to jointly learn visual and textual representations from large-scale data, and utilizes them to answer VQA questions. Specifically, we consider 8-layer ViLBERT implementation available at the link `https://github.com/jiasenlu/vilbert_beta`. On VQA train splits, we train the model for a maximum of 25 epochs and use early-stopping based on the validation performance. We use an initial learning rate of $3e^{-5}$ and use a linear decay learning rate schedule with warm up. We train on 8 Tesla V100 GPUs with a total batch size of 512.

Contrastive Learning using ViLBERT: In implementing the contrastive loss functions, we randomly sample negatives from the mini-batch for computational efficiency (similar to [AGA20b]). We sampled 64 negatives from each batch for both Sum-H and Max-H losses and fine-tune the margin parameters based on development split.

Multi-Task Learning using ViLBERT: We present our multi-task learning (MTL) architecture in Figure 6.10. The shared layers of ViLBERT constitute transformer blocks (TRM) and co-attentional transformer layers (Co-TRM) [LBP19a]. The weights for the task-specific layers are randomly initialized, whereas the shared layers are initialized with weights pre-trained on 3.3 million image-caption pairs from Conceptual Captions dataset [SDG18b]. We use a binary cross-entropy loss for all the auxiliary tasks GQA [HM19b], visual common sense reasoning (VCR) [ZBF19], and referring expression recognition (REF) [CMB18]. We considered RefCOCOg [MHT16] dataset for REF task. We optimize each task alternatively in mini-batches based on a mixing ratio and employ early-stopping based on the validation performance. In all our contrastive learning and multi-task learning experiments, we use an initial learning rate of 4e-5, and use a linear decay learning rate schedule with warm up. We train on 4 RTX 2080 GPUs with a total batch size of 256.

Transfer Learning using ViLBERT: In addition to the contrastive learning and MTL based adaptation results presented in Section 4 of main paper, we also explore transfer learning (TL) based models. Specifically, we first pre-train ViLBERT on auxiliary tasks, in contrast to joint training in MTL, and then fine-tune it on VQA train split. As shown in Table 6.13, we did not find any significant improvement in model's performance on CrossVQA.

6.7.6 More Details on CrossVQA

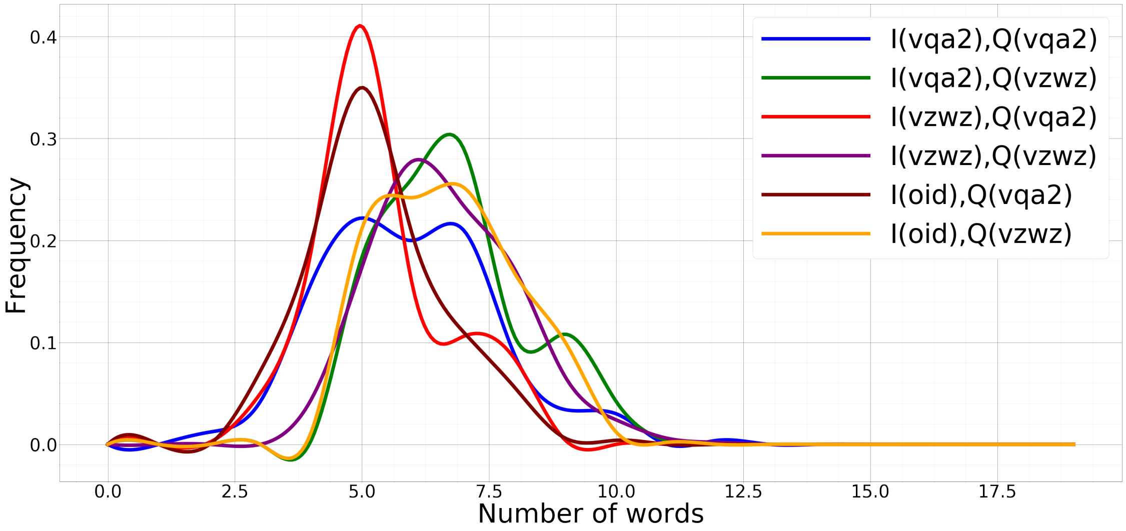


Figure 6.11: Question length distribution for all the six CrossVQA splits.

In addition to the statistics presented in the Section 4 of the main paper, we present additional details of our CrossVQA splits. Figure 6.12 and Figure 6.13 show a word cloud plot for the majority questions and answers across all the six splits. A variety of objects and answers can be seen in the plots, suggesting that our splits are diverse. Moreover, the relative frequency of the most frequent spatial relationships across all the six splits in Figure 6.14 show that CrossVQA comprises of rich and diverse spatial relationships. Figure 6.11 shows question length distribution of all the six splits. As we expected, we find that splits with VizWiz style questions, i.e. $\langle I_{vqa2}, QA_{vzwz}\rangle$, $\langle I_{vzwz}, QA_{vzwz}\rangle$, and $\langle I_{oid}, QA_{vzwz}\rangle$ contain more words in the question on average than other splits in CrossVQA.

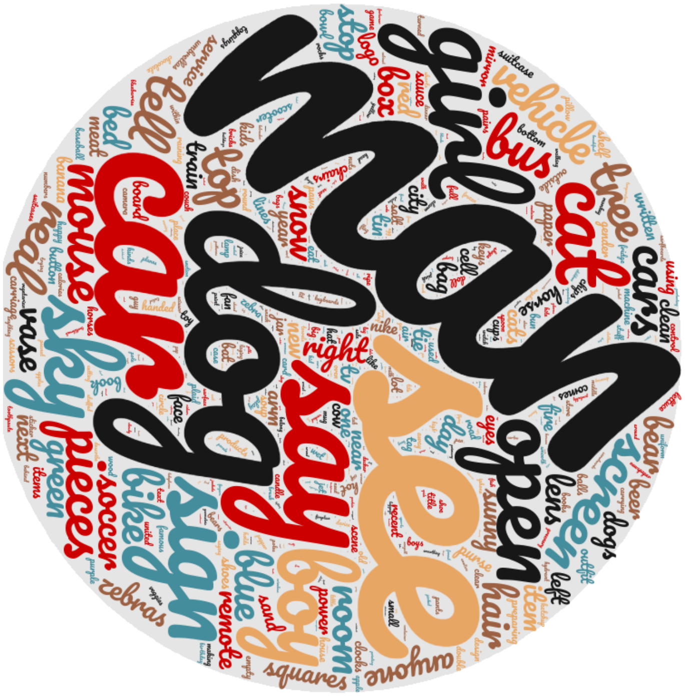


Figure 6.12: Wordcloud for questions

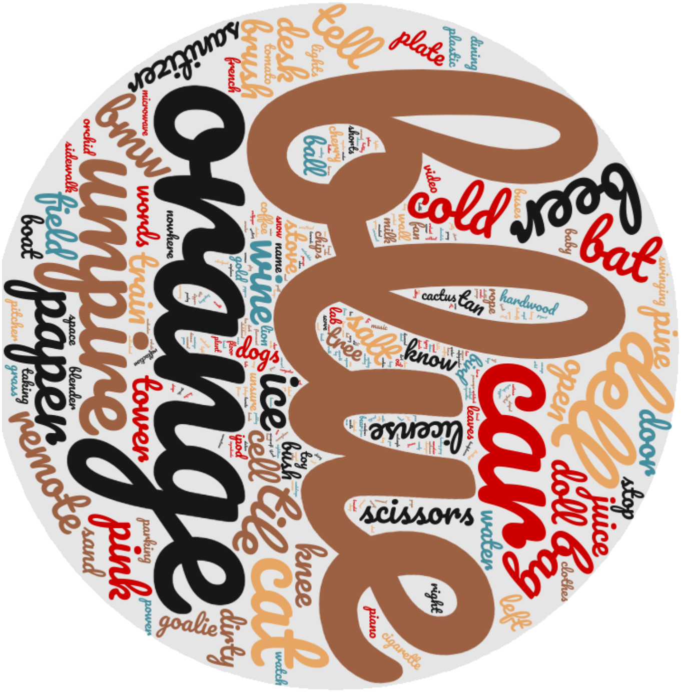


Figure 6.13: Wordcloud for answers across all the CrossVQA splits.

Model	Pre-train?	B1	B4	M	R	S	C	QS	QI	AS	AI	OC
										(0−100)	(0−100)	(0−10)
IC2Q	✗	30.42	4.44	9.42	-	-	0.27	11.37	34.76	-	-	-
V-IC2Q	✗	35.40	10.78	13.35	-	-	0.42	12.97	38.32	-	-	-
MI-IC2Q	✗	47.40	14.49	18.35	40.27	-	0.86	26.06	52.11	-	-	-
Ours (IC2QA)	✗	55.77	27.54	22.18	49.60	21.80	0.98	27.00	53.90	2.80	11.15	2.78
Ours (IC2QA)	✓	61.34	32.01	29.09	52.18	26.03	1.15	27.94	57.00	3.79	15.00	3.12
IA2Q	✗	32.43	6.23	11.21	-	-	0.36	-	-	-	-	-
V-IA2Q	✗	36.91	6.25	12.39	-	-	0.36	-	-	-	-	-
MI-IA2Q	✗	48.09	15.17	18.78	49.10	-	0.92	-	-	-	-	-
Ours (IA2QA)	✗	57.12	29.00	24.16	51.13	23.69	1.02	27.20	54.09	2.90	11.20	3.02
Ours (IA2QA)	✓	63.00	34.82	30.05	55.00	27.18	1.18	28.90	58.11	3.89	16.01	3.18
Ours (IÃ2QÃ)	✓	66.02	37.15	32.00	58.16	30.62	1.20	29.10	61.09	4.96	18.89	4.56
Ours (IÃC2QÃ)	✓	75.34	42.09	**41.52**	**69.41**	38.60	1.40	33.00	80.50	22.09	39.80	4.98
Ours (IÃP2QÃ)	✓	**79.52**	**44.74**	41.01	68.20	**39.87**	**1.54**	**33.12**	**82.50**	**23.50**	**39.86**	**5.74**

Table 6.5: Performance of our VQAG model against the baselines using the metrics BLEU-1 (B1), BLEU-4 (B4), METEOR (M), ROUGE-L (R), SPICE (S), CIDEr (C), Question generative strength (QS) and inventiveness (QI), answer generative strength (AS) and inventiveness (AI), and oracle cider (OC). “Pre-train?” refers to whether or not we pre-train our VQAG on Conceptual 12M [CSD21].

Test Set	#Images	#Questions	Question	Unique
			Vocab	Answers
$\langle I_{vqa2}, QA_{vqa2} \rangle$	3000	8418	976	464
$\langle I_{vqa2}, QA_{vzwz} \rangle$	3000	8986	927	389
$\langle I_{vzwz}, QA_{vqa2} \rangle$	3000	8438	872	440
$\langle I_{vzwz}, QA_{vzwz} \rangle$	3000	3014	1004	325
$\langle I_{oid}, QA_{vqa2} \rangle$	3000	8986	963	332
$\langle I_{oid}, QA_{vzwz} \rangle$	3000	8986	982	427

Table 6.6: Statistics of CrossVQA before human validation.

Test Set	QR	AC	Categories
			with AC $< 30\%$
$\langle I_{vqa2}, QA_{vqa2} \rangle$	97.8	69.8	*count, time*
$\langle I_{vqa2}, QA_{vzwz} \rangle$	96.0	74.8	*count, time, spatial*
$\langle I_{vzwz}, QA_{vqa2} \rangle$	69.8	52.07	*time, food, spatial*
$\langle I_{vzwz}, QA_{vzwz} \rangle$	82.2	61.2	*food, spatial, attribute*
$\langle I_{oid}, QA_{vqa2} \rangle$	77.4	51.6	*count, time, attribute*
$\langle I_{oid}, QA_{vzwz} \rangle$	81.4	63.7	*count, time, spatial*

Table 6.7: Human Evaluation: question relevance (QR) and answer correctness (AC).

Q_A from	Q_B from	JSD unigram	JSD bigram
VQA2	VizWiz	0.57	0.59
$\langle I_{vqa2}, QA_{vqa2}\rangle$	VQA2	0.06	0.07
$\langle I_{vqa2}, QA_{vzwz}\rangle$	VizWiz	0.09	0.08
$\langle I_{vzwz}, QA_{vqa2}\rangle$	VQA2	0.11	0.09
$\langle I_{vzwz}, QA_{vzwz}\rangle$	VizWiz	0.06	0.07

Table 6.8: Comparison of question distribution of source and the generated datasets measured using the Jensen-Shannon (JSD) divergence

Model	vqa2,vqa2	vqa2,vzwz	vzwz,vqa2	oid,vqa2
LXMERT	60.1	50.5	25.0	38.6
VisualBERT	58.1	55.1	21.4	43.6
ViLBERT(VB)	62.5	57.8	26.6	44.8
VB+Sum-H	62.8	57.8	26.9	43.9
VB+Max-H	64.1	58.0	26.9	42.8
VB+GQA	65.3	57.8	25.7	40.4
VB+RER	63.0	58.1	27.2	44.0
VB+VCR	61.0	54.3	24.1	39.6

Table 6.9: Performance on human-validated CrossVQA test sets with VQA2 images or VQA2-style questions for (i) the state-of-the-art models (top three rows) and (ii) ViLBERT (VB) with contrastive (Sum-H, Max-H) and multi-task (GQA, RER, VCR) losses.

Categories	**Examples**
`Count`	0, 1, 2, 30, 40, 200, many, lot, very
`Binary`	yes, no
`Predicate`	on ground, on plate
`Material`	wood, plastic, concrete, oak, plaid
`Time`	afternoon, sunset, morning, spring
`Color`	white, blue, red, black
`Attribute`	sunny, male, winter, stripes, open
`Object`	frisbee, water, grass, skateboard, phone
`Stuff`	sky
`Food`	vegetables, tomato, salad, milk, dessert
`Shape`	rectangle, triangle, oval, round
`Other`	nothing, english, electricity, united
`Location`	living room, beach, ocean, mountains
`Animal`	cat, dog, zebras, person, police
`Spatial`	right, left, front, downhill, north
`Activity`	skateboarding, standing, playing wii

Table 6.10: Answer categories in our VQAG Model

Examples of Invented Questions	**Examples of Invented Answers**
Q1: What hand is the man using to write with?	{at least 10 years, above doorway, inside the baggage, behind red car, towards bottom left side, dirt bikes, fishing boats, fork and sharp knife, riding big elephants, right side of road}
Q2: Are most of the lights on or off in the living room?	
Q3: Will this woman be drinking beer?	
Q4: What is the number on the front side of the bike?	
Q5: In this scene how many sheep can be clearly seen?	
Q6: What is the purpose of the number on the yellow board?	
Q7: Which sheep is the older in the picture?	
Q8: Is the fire hydrant old or new?	
Q9: What is the first letter of the word on the blue sign?	
Q10: What is the name of the logo on top of the keyboard?	

Table 6.11: Examples of unseen questions and answers invented by our VQAG Model

	Seen+Unseen		**Seen**		**Unseen**	
	QR	AC	QR	AC	QR	AC
VQA2.0 val split	90.6	61.7	93.2	74.7	84.6	58.8
VizWiz val split	91.2	54.2	92.8	59.7	86.1	48.3
OpenImages val split	88.8	57.0	89.1	60.9	85.7	49.1

Table 6.12: Comparison of question relevance (QR) and answer correctness (AC) on in-domain val splits (VQA2.0) and out-of-domain splits (VizWiz, OpenImages).

Model	vqa2,vqa2	vqa2,vzwz	vzwz,vqa2	oid,vqa2
VB	62.5	57.8	26.6	44.8
VB+TL(GQA)	59.3	57.9	26.0	42.1
VB+TL(REF)	58.4	54.2	24.1	40.2
VB+TL(VCR)	59.7	56.3	25.0	41.4

Table 6.13: Adaptation Results on CrossVQA with Transfer Learning

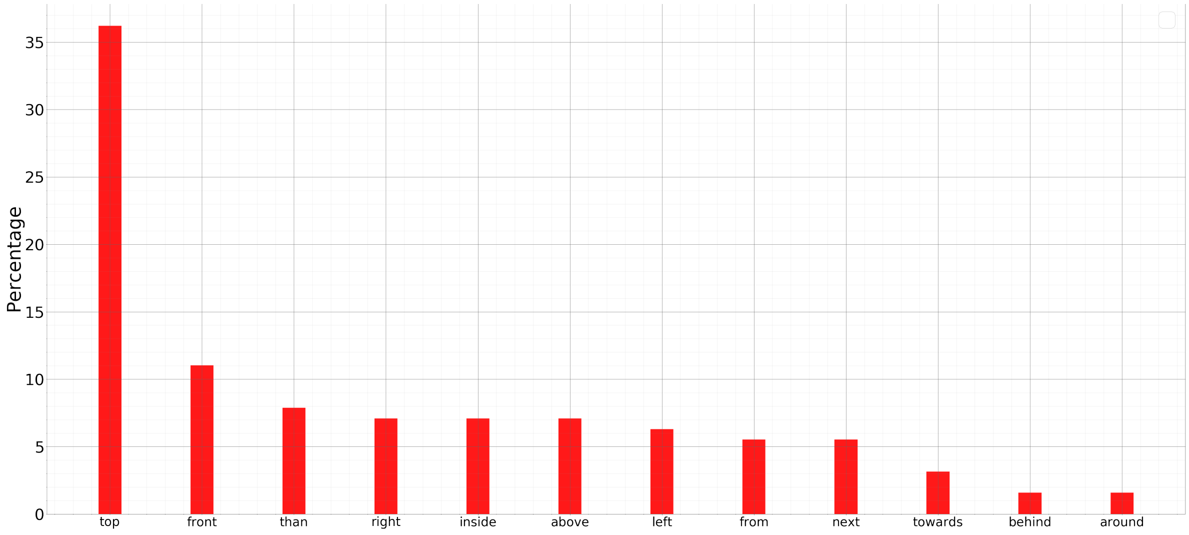


Figure 6.14: Relative frequency of the most frequent spatial relationships in CrossVQA.

CHAPTER 7

Conclusion

In this thesis, we demonstrate novel methods and algorithms to effectively gain human trust in vision and language reasoning models by generating adaptive and human understandable explanations and also by improving transparency, interpretability, faithfulness, and robustness of the existing deep learning models. We presented X-ToM – a new framework for Explainable AI (XAI) and human trust evaluation based on the Theory-of-Mind (ToM). X-ToM generates explanations in a dialog by explicitly modeling, learning, and inferring three mental states based on And-Or Graphs – namely, machine's mind, human's mind as inferred by the machine, and machine's mind as inferred by the human. This allows for a principled formulation of human trust in the machine. For the task of visual recognition, we proposed a novel, collaborative task-solving game that can be used for collecting training data and thus learning the three mental states, as well as a testbed for quantitative evaluation of explainable vision systems. We demonstrated the superiority of X-ToM in gaining human trust relative to baselines. We also introduced a new explainable AI (XAI) framework based on fault-lines. We argue that due to their conceptual and counterfactual nature, fault-line based explanations are lucid, clear and easy for humans to understand. We proposed a new method to automatically mine explainable concepts from a given training dataset and to derive fault-line explanations. Using qualitative and quantitative evaluation metrics, we demonstrated that fault-lines significantly outperform baselines in improving human understanding of the underlying classification model.

We further evaluate the existing deep learning models such as Transformer, Compositional Modular Networks in terms of their ability to provide interpretable visual and language representations and their ability to provide robust predictions to out-of-distribution samples. Our work shows that current datasets and models for vision and language grounding tasks such as visual question answering and visual referring expression recognition tasks, fail to make effective use of linguistic structure. Although our proposed models are slightly more robust than existing models, there is still significant scope for improvement. We hope that our newly introduced adversarial test splits will foster more research in this area.

We find evidence that that the state-of-the-art end-to-end modular network (NMN) implementations - although provide high model interpretability with their transparent, hierarchical and semantically motivated architecture - require a large amount of training data and are less effective in generalizing to unseen but known language constructs. We also demonstrate that explicitly conditioning neural modules on the language guidance through adaptive convolutions improve their grounding and generalization abilities, achieving a new state-of-the-art results on the visual question answering and visual referring expression recognition tasks. Our analysis on CLOSURE, CLEVR-Ref+ and a new compositional and contrastive split C3-Ref+ demonstrate that our proposed method enhances NMN' ability in adaptively selecting and exploiting informative visiolinguistic relationships.

Finally, we present a step toward scalable and systematic evaluation of visual question answering systems. Key to our approach is an accurate and controllable VQAG module that is capable of generating disentangled distribution shifts. We generate CrossVQA benchmarks, a collection of test splits based on VQA2, VizWiz, and Open Images datasets. We validate their utility by showing that existing VQA models struggle to perform well in this evaluation scenario and identifying the image distribution mismatch as the main factor.

REFERENCES

[AAA17] Shivali Agarwal, Vishalaksh Aggarwal, Arjun R Akula, Gargi Banerjee Dasgupta, and Giriprasad Sridhara. “Automatic problem extraction and analysis from unstructured text in IT tickets.” *IBM Journal of Research and Development*, **61**(1):4–41, 2017.

[AAD18] Shivali Agarwal, Arjun R Akula, Gaargi B Dasgupta, Shripad J Nadgowda, and Tapan K Nayak. “Structured representation and classification of noisy and unstructured tickets in service delivery.”, October 9 2018. US Patent 10,095,779.

[AAL15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. “Vqa: Visual question answering.” In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2425–2433, 2015.

[ABP18] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. “Don’t just assume; look and answer: Overcoming priors for visual question answering.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4971–4980, 2018.

[ACG21] Arjun R Akula, Beer Changpinyo, Boqing Gong, Piyush Sharma, Song-Chun Zhu, and Radu Soricut. “CrossVQA: Scalably Generating Benchmarks for Systematically Testing VQA Generalization.” 2021.

[ADE21] Arjun R Akula, Gargi B Dasgupta, Vijay Ekambaram, and Ramasuri Narayanam. “Measuring effective utilization of a service practitioner for ticket resolution via a wearable device.”, February 23 2021. US Patent 10,929,264.

[ADN18] Arjun R Akula, Gaargi B Dasgupta, and Tapan K Nayak. “Analyzing tickets using discourse cues in communication logs.”, September 4 2018. US Patent 10,067,983.

[ADW19] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. “nocaps: novel object captioning at scale.” In *ICCV*, 2019.

[AFJ16] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. “Spice: Semantic propositional image caption evaluation.” In *European Conference on Computer Vision*, pp. 382–398. Springer, 2016.

[AGA20a] Arjun R Akula, Spandana Gella, Yaser Al-Onaizan, Song-Chun Zhu, and Siva Reddy. “Words aren’t enough, their order matters: On the Robustness of Grounding Visual Referring Expressions.” *arXiv preprint arXiv:2005.01655*, 2020.

[AGA20b] Arjun R. Akula, Spandana Gella, Yaser Al-Onaizan, Song-Chun Zhu, and Siva Reddy. "Words aren't enough, their order matters: On the Robustness of Grounding Visual Referring Expressions." In *ACL*, 2020.

[AGW21] Arjun Akula, Spandana Gella, Keze Wang, Song-chun Zhu, and Siva Reddy. "Mind the Context: The Impact of Contextualization in Neural Module Networks for Grounding Visual Referring Expressions." In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6398–6416, 2021.

[AJ18] David Alvarez-Melis and Tommi S Jaakkola. "On the robustness of interpretability methods." *arXiv preprint arXiv:1806.08049*, 2018.

[AJC21] Arjun Akula, Varun Jampani, Soravit Changpinyo, and Song-Chun Zhu. "Robust Visual Reasoning via Language Guided Neural Module Networks." *Advances in Neural Information Processing Systems*, **34**, 2021.

[AK12] M Gethsiyal Augasta and Thangairulappan Kathirvalavakumar. "Reverse engineering the neural networks for rule extraction in classification problems." *Neural processing letters*, **35**(2):131–150, 2012.

[Aku15] Arjun R Akula. "A novel approach towards building a generic, portable and contextual nlidb system." *International Institute of Information Technology Hyderabad*, 2015.

[Ala17] N Alang. "Turns Out Algorithms Are Racist.[online] The New Republic.", 2017.

[ALC19] Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. "Fusion of detected objects in text for visual question answering." In *EMNLP-IJCNLP*, 2019.

[ALS19] Arjun R Akula, Changsong Liu, Sari Saba-Sadiya, Hongjing Lu, Sinisa Todorovic, Joyce Y Chai, and Song-Chun Zhu. "X-tom: Explaining with theory-of-mind for gaining justified human trust." *arXiv preprint arXiv:1909.06907*, 2019.

[ALT19] Arjun R Akula, Changsong Liu, Sinisa Todorovic, Joyce Y Chai, and Song-Chun Zhu. "Explainable AI as Collaborative Task Solving." In *CVPR Workshops*, pp. 91–94, 2019.

[ARD16a] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. "Learning to compose neural networks for question answering." *arXiv preprint arXiv:1601.01705*, 2016.

[ARD16b] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. "Neural module networks." In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 39–48, 2016.

[ASM13] Arjun Akula, Rajeev Sangal, and Radhika Mamidi. "A novel approach towards incorporating context processing capabilities in nlidb system." In *Proceedings of the sixth international joint conference on natural language processing*, pp. 1216–1222, 2013.

[ATC19] Arjun R Akula, Sinisa Todorovic, Joyce Y Chai, and Song-Chun Zhu. "Natural Language Interaction with Explainable AI Models." In *CVPR Workshops*, pp. 87–90, 2019.

[AWL21] Arjun R Akula, Keze Wang, Changsong Liu, Sari Saba-Sadiya, Hongjing Lu, Sinisa Todorovic, Joyce Chai, and Song-Chun Zhu. "CX-ToM: Counterfactual Explanations with Theory-of-Mind for Enhancing Human Trust in Image Recognition Models." *arXiv preprint arXiv:2109.01401*, 2021.

[AWZ20] Arjun R. Akula, Shuai Wang, and Song-Chun Zhu. "CoCoX: Generating Conceptual and Counterfactual Explanations via Fault-Lines." In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 2594–2601. AAAI Press, 2020.

[AZ19] Arjun R Akula and Song-Chun Zhu. "Visual discourse parsing." *ArXiv preprint*, **abs/1903.02252**, 2019.

[BB87] Dianne C Berry and Donald E Broadbent. "Explanation and verbalization in a computer-assisted search task." *The Quarterly Journal of Experimental Psychology*, **39**(4):585–609, 1987.

[BBM15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation." *PloS one*, **10**(7):e0130140, 2015.

[BC17] Or Biran and Courtenay Cotton. "Explanation and justification in machine learning: A survey." In *IJCAI-17 workshop on explainable AI (XAI)*, volume 8, p. 1, 2017.

[BDH18] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. "AI fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias." *arXiv preprint arXiv:1810.01943*, 2018.

[BL05] Satanjeev Banerjee and Alon Lavie. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments." In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

[BLG19] Nilavra Bhattacharya, Qing Li, and Danna Gurari. "Why Does a Visual Question Have Different Answers?" In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 4270–4279. IEEE, 2019.

[Bor16] Aaron M Bornstein. "Is Artificial Intelligence Permanently Inscrutable?" 2016.

[BRH17] A Bivens, H Ramasamy, LM Herger, WJ Rippon, CA Fonseca, W Pointer, BM Belgodere, WH Cornejo, MJ Frissora, V Ramakrishna, et al. "Cognitive and Contextual Analytics for IT Services." 2017.

[Bri] Chris Brinton. "A Framework for Explanation of Machine Learning Decisions." In *IJCAI-17 Workshop on Explainable AI (XAI)*, p. 14.

[BRW20] Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. "Beat the AI: Investigating Adversarial Human Annotation for Reading Comprehension." *Transactions of the Association for Computational Linguistics*, **8**:662–678, 2020.

[BSB20] Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. "Adversarial Filters of Dataset Biases." In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1078–1088. PMLR, 2020.

[BT09] Amir Beck and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." *SIAM journal on imaging sciences*, **2**(1):183–202, 2009.

[BVO19] Dzmitry Bahdanau, Harm de Vries, Timothy J O'Donnell, Shikhar Murty, Philippe Beaudoin, Yoshua Bengio, and Aaron Courville. "CLOSURE: Assessing Systematic Generalization of CLEVR Models." *arXiv preprint arXiv:1912.05783*, 2019.

[CBM18] Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. "Using syntax to ground referring expressions in natural images." In *AAAI Conference on Artificial Intelligence*, 2018.

[CBP15] Eric T Chancey, James P Bliss, Alexandra B Proaps, and Poornima Madhavan. "The role of trust as a mediator between system characteristics and response behaviors." *Human factors*, **57**(6):947–958, 2015.

[CBS17] Cary Champlin, David Bell, and Celina Schocken. "AI medicine comes to Africa's rural clinics." *IEEE Spectrum*, **54**(5):42–48, 2017.

[CHS18a] Wei-Lun Chao, Hexiang Hu, and Fei Sha. "Being Negative but Constructively: Lessons Learnt from Creating Better Visual Question Answering Datasets." In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 431–441, New Orleans, Louisiana, 2018. Association for Computational Linguistics.

[CHS18b] Wei-Lun Chao, Hexiang Hu, and Fei Sha. "Cross-dataset adaptation for visual question answering." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5716–5725, 2018.

[CKL19] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. "What Does BERT Look At? An Analysis of BERT's Attention." *arXiv preprint arXiv:1906.04341*, 2019.

[CMB18] Volkan Cirik, Louis-Philippe Morency, and Taylor Berg-Kirkpatrick. "Visual Referring Expression Recognition: What Do Systems Actually Learn?" In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pp. 781–787, 2018.

[CPS19] Soravit Changpinyo, Bo Pang, Piyush Sharma, and Radu Soricut. "Decoupled Box Proposal and Featurization with Ultrafine-Grained Semantic Labels Improve Image Captioning and Visual Question Answering." In *EMNLP-IJCNLP*, 2019.

[CS89] Herbert H Clark and Edward F Schaefer. "Contributing to discourse." *Cognitive science*, **13**(2):259–294, 1989.

[CSD21] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. "Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts." In *CVPR*, 2021.

[CW86] Herbert H Clark and Deanna Wilkes-Gibbs. "Referring as a collaborative process." *Cognition*, **22**(1):1–39, 1986.

[DA16] Sandra Devin and Rachid Alami. "An implemented theory of mind to improve human-robot shared plans execution." In *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*, pp. 319–326. IEEE, 2016.

[Dar13] Keith Darlington. "Aspects of intelligent systems explanation." *Universal Journal of Control and Automation*, **1**(2):40–51, 2013.

[DCL18] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. "Explanations based on the missing: Towards contrastive explanations with pertinent negatives." In *Advances in Neural Information Processing Systems*, pp. 592–603, 2018.

[DDP15] Amit Datta, Anupam Datta, Ariel D Procaccia, and Yair Zick. "Influence in Classification via Cooperative Game Theory." In *IJCAI*, pp. 511–517, 2015.

[DK17a] Finale Doshi-Velez and Been Kim. "A roadmap for a rigorous science of interpretability." *arXiv preprint arXiv:1702.08608*, **150**, 2017.

[DK17b] Finale Doshi-Velez and Been Kim. "Towards a rigorous science of interpretable machine learning." *arXiv preprint arXiv:1702.08608*, 2017.

[DNA14] Gargi B Dasgupta, Tapan K Nayak, Arjun R Akula, Shivali Agarwal, and Shripad J Nadgowda. "Towards auto-remediation in services delivery: Context-based classification of noisy and unstructured tickets." In *International Conference on Service-Oriented Computing*, pp. 478–485. Springer, 2014.

[DQX17] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. "Deformable convolutional networks." In *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017.

[EBC09] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Visualizing higher-layer features of a deep network." *Technical report, University of Montreal*, **1341**(3):1, 2009.

[FFK18] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. "VSE++: Improving Visual-Semantic Embeddings with Hard Negatives." In *British Machine Vision Conference*, p. 12, 2018.

[FRD18] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. "Model class reliance: Variable importance measures for any machine learning model class, from the "Rashomon" perspective." *arXiv preprint arXiv:1801.01489*, 2018.

[Fri01] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine." *Annals of statistics*, pp. 1189–1232, 2001.

[FV17] Ruth C Fong and Andrea Vedaldi. "Interpretable explanations of black boxes by meaningful perturbation." In *Proceedings of the IEEE international conference on computer vision*, pp. 3429–3437, 2017.

[GAB20] Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. "Evaluating nlp models via contrast sets." *arXiv preprint arXiv:2004.02709*, 2020.

[GAD16] Abhirut Gupta, Arjun Akula, Gargi Dasgupta, Pooja Aggarwal, and Prateeti Mohapatra. "Desire: Deep semantic understanding and retrieval for technical support services." In *International Conference on Service-Oriented Computing*, pp. 207–210. Springer, 2016.

[GAM12] Abhijeet Gupta, Arjun Akula, Deepak Malladi, Puneeth Kukkadapu, Vinay Ainavolu, and Rajeev Sangal. "A novel approach towards building a portable nlidb system using the computational paninian grammar framework." In *2012 International Conference on Asian Language Processing*, pp. 93–96. IEEE, 2012.

[GF17] Bryce Goodman and Seth Flaxman. "European Union regulations on algorithmic decision-making and a "right to explanation"." *AI Magazine*, **38**(3):50–57, 2017.

[GKS17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. "Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017.

[GL15] Yaroslav Ganin and Victor Lempitsky. "Unsupervised domain adaptation by backpropagation." In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

[GLS18] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. "Vizwiz grand challenge: Answering visual questions from blind people." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617, 2018.

[Gol12] Alvin I Goldman. *Theory of Mind*. The Oxford handbook of philosophy of cognitive science, 2012.

[GPC16] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs." *Jama*, **316**(22):2402–2410, 2016.

[GPL03] Russ Greiner, B Poulin, Paul Lu, J Anvik, Z Lu, Cam Macdonell, David Wishart, Roman Eisner, and Duane Szafron. "Explaining Naive Bayes Classifications." 2003.

[GS01] Frédéric Gosselin and Philippe G Schyns. "Bubbles: a technique to reveal the use of information in recognition tasks." *Vision research*, **41**(17):2261–2271, 2001.

[GSK17] Spandana Gella, Rico Sennrich, Frank Keller, and Mirella Lapata. "Image Pivoting for Learning Multilingual Multimodal Representations." In *Empirical Methods in Natural Language Processing*, pp. 2839–2845, September 2017.

[GSS12] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. "Geodesic flow kernel for unsupervised domain adaptation." In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2066–2073. IEEE, 2012.

[GSS14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572*, 2014.

[GWD14] Alex Graves, Greg Wayne, and Ivo Danihelka. "Neural Turing Machines." *arXiv preprint arXiv:1410.5401*, 2014.

[GWE19] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. "Counterfactual Visual Explanations." In *ICML 2019*, 2019.

[GWK19] Amirata Ghorbani, James Wexler, and Been Kim. "Automating Interpretability: Discovering and Testing Visual Concepts Learned by Neural Networks." *arXiv preprint arXiv:1902.03129*, 2019.

[GX12] Yuhong Guo and Min Xiao. "Cross language text classification via subspace co-regularized multi-view learning." *arXiv preprint arXiv:1206.6481*, 2012.

[Hal19] Patrick Hall. "Guidelines for Responsible and Human-Centered Use of Explainable Machine Learning." *arXiv preprint arXiv:1906.03533*, 2019.

[HAR16] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. "Generating visual explanations." In *European Conference on Computer Vision*, pp. 3–19. Springer, 2016.

[HAR17] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. "Learning to reason: End-to-end module networks for visual question answering." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 804–813, 2017.

[HHB10] Robert R Hoffman, Peter A Hancock, and Jeffrey M Bradshaw. "Metrics, Metrics, Metrics, Part 2: Universal Metrics?" *IEEE Intelligent Systems*, **25**(6):93–97, 2010.

[HHD18] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. "Generating Counterfactual Explanations with Natural Language." In *ICML Workshop on Human Interpretability in Machine Learning*, 2018.

[Hil90] Denis J Hilton. "Conversational processes and causal explanation." *Psychological Bulletin*, **107**(1):65, 1990.

[HK17] Robert R Hoffman and Gary Klein. "Explaining explanation, part 1: theoretical foundations." *IEEE Intelligent Systems*, **32**(3):68–73, 2017.

[HK18] Alex Hernández-García and Peter König. "Do deep nets really need weight decay and dropout?" *arXiv preprint arXiv:1802.07042*, 2018.

[HM19a] John Hewitt and Christopher D Manning. "A structural probe for finding syntax in word representations." In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4129–4138, 2019.

[HM19b] Drew A Hudson and Christopher D Manning. "Gqa: A new dataset for real-world visual reasoning and compositional question answering." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6700–6709, 2019.

[HM19c] Drew A Hudson and Christopher D Manning. "Learning by abstraction: The neural state machine." In *Proceedings of NeurIPS*, 2019.

[HMK18] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. "Metrics for Explainable AI: Challenges and Prospects." *arXiv preprint arXiv:1812.04608*, 2018.

[Hof17a] Robert R Hoffman. "A taxonomy of emergent trusting in the human–machine relationship." *Cognitive systems engineering: The future for a changing world*, pp. 137–163, 2017.

[Hof17b] R.R. Hoffman. "A Taxonomy of Emergent Trusting in the Human–Machine Relationship." *Cognitive systems engineering: The future for a changing world*, 2017.

[HRA17] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. "Modeling relationships in referential expressions with compositional modular networks." In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1115–1124, 2017.

[HRD19] Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. "Language-conditioned graph networks for relational reasoning." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10294–10303, 2019.

[HTF01] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531*, 2015.

[HXR16] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. "Natural language object retrieval." In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4555–4564, 2016.

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[JDT16] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. "Dynamic filter networks." In *Advances in neural information processing systems*, pp. 667–675, 2016.

[JE10] Sam Johnson and Mark Everingham. "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation." In *BMVC*, volume 2, p. 5, 2010.

[JHM17a] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning." In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[JHM17b] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. "Inferring and Executing Programs for Visual Reasoning." In *ICCV*, 2017.

[JJV16] Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. “Revisiting visual question answering baselines.” In *Proceedings of ECCV*, 2016.

[JLL19] Da-Cheng Juan, Chun-Ta Lu, Zhen Li, Futang Peng, Aleksei Timofeev, Yi-Ting Chen, Yaxi Gao, Tom Duerig, Andrew Tomkins, and Sujith Ravi. “Graph-rise: Graph-regularized image semantic embedding.” *arXiv preprint arXiv:1902.10814*, 2019.

[JW19] Sarthak Jain and Byron C. Wallace. “Attention is not Explanation.” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2019.

[JZS17] Unnat Jain, Ziyu Zhang, and Alexander G Schwing. “Creativity: Generating diverse questions using variational autoencoders.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6485–6494, 2017.

[KB15] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization.” *International Conference on Learning Representations (ICLR)*, 2015.

[KBF19] Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. “Information maximizing visual question generation.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2008–2018, 2019.

[KHL20] Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. “Learning The Difference That Makes A Difference With Counterfactually-Augmented Data.” In *International Conference on Learning Representations*, 2020.

[KJF15] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. “Visualizing and understanding recurrent networks.” *arXiv preprint arXiv:1506.02078*, 2015.

[KOM14] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. “Referitgame: Referring to objects in photographs of natural scenes.” In *Empirical methods in natural language processing (EMNLP)*, pp. 787–798, 2014.

[KRA18] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. “The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale.” *arXiv preprint arXiv:1811.00982*, 2018.

[KRS14] Been Kim, Cynthia Rudin, and Julie A Shah. “The bayesian case model: A generative approach for case-based reasoning and prototype classification.” In *Advances in Neural Information Processing Systems*, pp. 1952–1960, 2014.

[KSD15] Been Kim, Julie A Shah, and Finale Doshi-Velez. “Mind the gap: A generative approach to interpretable feature selection and extraction.” In *Advances in Neural Information Processing Systems*, pp. 2260–2268, 2015.

[KSZ14] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. “Unifying visual-semantic embeddings with multimodal neural language models.” *arXiv preprint arXiv:1411.2539*, 2014.

[KT81] Daniel Kahneman and Amos Tversky. “The simulation heuristic.” Technical report, STANFORD UNIV CA DEPT OF PSYCHOLOGY, 1981.

[KWG18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. “Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV).” In *International Conference on Machine Learning*, pp. 2673–2682, 2018.

[KZG17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. “Visual Genome: Connecting language and vision using crowdsourced dense image annotations.” *International Journal of Computer Vision*, **123**(1):32–73, 2017.

[LB17] Brenden M Lake and Marco Baroni. “Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks.” *arXiv preprint arXiv:1711.00350*, 2017.

[LBJ16] Tao Lei, Regina Barzilay, and Tommi Jaakkola. “Rationalizing neural predictions.” *arXiv preprint arXiv:1606.04155*, 2016.

[LBP19a] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. “Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks.” In *Advances in Neural Information Processing Systems*, pp. 13–23, 2019.

[LBP19b] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. “ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks.” In *NeurIPS*, 2019.

[LCW17] Joseph B Lyons, Matthew A Clark, Alan R Wagner, and Matthew J Schuelke. “Certifiable Trust in Autonomous Systems: Making the Intractable Tangible.” *AI Magazine*, **38**(3), 2017.

[LHW13] Bo Li, Wenze Hu, Tianfu Wu, and Song-Chun Zhu. “Modeling occlusion by discriminative and-or structures.” In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2560–2567, 2013.

[Lin04] Chin-Yew Lin. “Rouge: A package for automatic evaluation of summaries.” In *Text summarization branches out*, pp. 74–81, 2004.

[Lip90] Peter Lipton. “Contrastive explanation.” *Royal Institute of Philosophy Supplements*, **27**:247–266, 1990.

[Lip16] Zachary C Lipton. “The mythos of model interpretability.” In *ICML Workshop on Human Interpretability in Machine Learning*, 2016.

[LLB19] Runtao Liu, Chenxi Liu, Yutong Bai, and Alan L. Yuille. “CLEVR-Ref+: Diagnosing Visual Reasoning With Referring Expressions.” In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pp. 4185–4194, 2019.

[LLS15] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. “Multi-task sequence to sequence learning.” *arXiv preprint arXiv:1511.06114*, 2015.

[LMB14a] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. “Microsoft coco: Common objects in context.” In *Proceedings of the European Conference on Computer Vision*, 2014.

[LMB14b] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. “Microsoft COCO: Common Objects in Context.” In *ECCV*, 2014.

[LNH09] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. “Learning to detect unseen object classes by between-class attribute transfer.” In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 951–958. IEEE, 2009.

[Lom06] Tania Lombrozo. “The structure and function of explanations.” *Trends in cognitive sciences*, **10**(10):464–470, 2006.

[LYS16] Changsong Liu, Shaohua Yang, Sari Saba-Sadiya, Nishant Shukla, Yunzhong He, Song-Chun Zhu, and Joyce Chai. “Jointly learning grounded task structures from language instruction and visual demonstration.” In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1482–1492, 2016.

[LYY19] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. “Visual-BERT: A Simple and Performant Baseline for Vision and Language.”, 2019.

[MBT19] Travis Mandel, Jahnu Best, Randall H Tanaka, Hiram Temple, Chansen Haili, Kayla Schlectinger, and Roy Szeto. “Let’s Keep It Safe: Designing User Interfaces that Allow Everyone to Contribute to AI Safety.” *arXiv preprint arXiv:1907.04446*, 2019.

[MFF17] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. “Universal adversarial perturbations.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, 2017.

[MH08] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” *Journal of machine learning research*, **9**(Nov):2579–2605, 2008.

[MHT16] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. “Generation and comprehension of unambiguous object descriptions.” In *IEEE conference on computer vision and pattern recognition*, pp. 11–20, 2016.

[Mil18a] Tim Miller. “Explanation in artificial intelligence: Insights from the social sciences.” *Artificial Intelligence*, 2018.

[Mil18b] Tim Miller. “Explanation in artificial intelligence: Insights from the social sciences.” *Artificial Intelligence*, 2018.

[MKS13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. “Playing Atari with Deep Reinforcement Learning.” *CoRR*, **abs/1312.5602**, 2013.

[MMD16] Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. “Generating natural questions about an image.” *arXiv preprint arXiv:1603.06059*, 2016.

[Mol19] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2019.

[NMD16] Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. “Modeling Context Between Objects for Referring Expression Understanding.” In *European Conference on Computer Vision*, pp. 792–807, 2016.

[NWD19] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. “Adversarial nli: A new benchmark for natural language understanding.” *arXiv preprint arXiv:1910.14599*, 2019.

[PAS13] Vasu Pulijala, Arjun R Akula, and Azeemuddin Syed. “A web-based virtual laboratory for electromagnetic theory.” In *2013 IEEE Fifth International Conference on Technology for Education (t4e 2013)*, pp. 13–18. IEEE, 2013.

[PGG18] Dino Pedreschi, Fosca Giannotti, Riccardo Guidotti, Anna Monreale, Luca Pappalardo, Salvatore Ruggieri, and Franco Turini. “Open the black box data-driven explanation of black box decision systems.” *arXiv preprint arXiv:1806.09936*, 2018.

[PNZ18] Seyoung Park, Bruce Xiaohan Nie, and Song-Chun Zhu. “Attribute and-or grammar for joint parsing of human attributes, part and pose.” *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 1555–1569, 2018.

[PPA18] Antonio Polino, Razvan Pascanu, and Dan Alistarh. “Model compression via distillation and quantization.” *arXiv preprint arXiv:1802.05668*, 2018.

[PRA15] Ashish Palakurthi, SM Ruthu, Arjun Akula, and Radhika Mamidi. “Classification of attributes in a natural language query into different SQL clauses.” In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pp. 497–506, 2015.

[PRW02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. “BLEU: a method for automatic evaluation of machine translation.” In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

[PSN19] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. “Competence-based Curriculum Learning for Neural Machine Translation.” *Proceedings of the 2019 Conference of the North*, 2019.

[PSV18] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. “FiLM: Visual Reasoning with a General Conditioning Layer.” In *AAAI*, 2018.

[PUC19] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. “Connecting Vision and Language with Localized Narratives.” *arXiv preprint arXiv:1912.03098*, 2019.

[PW78] David Premack and Guy Woodruff. “Does the chimpanzee have a theory of mind?” *Behavioral and brain sciences*, **1**(4):515–526, 1978.

[QWL15] Hang Qi, Tianfu Wu, Mun-Wai Lee, and Song-Chun Zhu. “A restricted visual turing test for deep scene and event understanding.” *ArXiv preprint*, **abs/1512.01715**, 2015.

[RAR16] RS Ramprasaath, D Abhishek, V Ramakrishna, C Michael, P Devi, and B Dhruv. “Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization.” *CVPR 2016*, 2016.

[RDS15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. “Imagenet large scale visual recognition challenge.” *International journal of computer vision*, **115**(3):211–252, 2015.

[RHG16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. “Faster r-cnn: Towards real-time object detection with region proposal networks.” *IEEE transactions on pattern analysis and machine intelligence*, **39**(6):1137–1149, 2016.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Why should i trust you?: Explaining the predictions of any classifier.” In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.

[RSG18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Anchors: High-precision model-agnostic explanations.” In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[SBG20] Sanjay Subramanian, Ben Bogin, Nitish Gupta, Tomer Wolfson, Sameer Singh, Jonathan Berant, and Matt Gardner. "Obtaining Faithful Interpretations from Compositional Neural Networks." *arXiv preprint arXiv:2005.00724*, 2020.

[SCD17a] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-cam: Visual explanations from deep networks via gradient-based localization." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

[SCD17b] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *ICCV*, 2017.

[SDG18a] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning." In *Association for Computational Linguistics*, pp. 2556–2565, 2018.

[SDG18b] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. "Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning." In *ACL*, 2018.

[SGH16] Hendrik Strobelt, Sebastian Gehrmann, Bernd Huber, Hanspeter Pfister, and Alexander M Rush. "Visual analysis of hidden state dynamics in recurrent neural networks." *arXiv preprint arXiv:1606.07461*, 2016.

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning important features through propagating activation differences." In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3145–3153. JMLR. org, 2017.

[SGL03] Duane Szafron, Russell Greiner, Paul Lu, David Wishart, Cam MacDonell, John Anvik, Brett Poulin, Zhiyong Lu, and Roman Eisner. "Explaining naïve Bayes classifications." *TR03-09, Department of Computing Science, University of Alberta*, 2003.

[She17] Raymond Ka-Man Sheh. "" Why Did You Do That?" Explainable Intelligent Robots." In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[SJS19] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. "Pixel-adaptive convolutional neural networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11166–11175, 2019.

[SK11] Erik Štrumbelj and Igor Kononenko. "A general method for visualizing and explaining black-box regression models." In *International Conference on Adaptive and Natural Computing Algorithms*, pp. 21–30. Springer, 2011.

[SM18] Raymond Sheh and Isaac Monteath. “Defining Explainable AI for Requirements Analysis.” *KI-Künstliche Intelligenz*, **32**(4):261–266, 2018.

[ST01] Makoto Sato and Hiroshi Tsukimoto. “Rule extraction from neural networks via decision tree induction.” In *IJCNN’01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, volume 3, pp. 1870–1875. IEEE, 2001.

[STK17] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. “Smoothgrad: removing noise by adding noise.” *arXiv preprint arXiv:1706.03825*, 2017.

[STT11] Ruslan Salakhutdinov, Antonio Torralba, and Josh Tenenbaum. “Learning to share visual appearance for multiclass object detection.” In *CVPR 2011*, pp. 1481–1488. IEEE, 2011.

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. “Axiomatic attribution for deep networks.” *34th International Conference on Machine Learning*, 2017.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to sequence learning with neural networks.” In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[SW18] Keng Siau and Weiyu Wang. “Building trust in artificial intelligence, machine learning, and robotics.” *Cutter Business Technology Journal*, **31**(2):47–53, 2018.

[SWS17] Austin Stone, Huayan Wang, Michael Stark, Yi Liu, D Scott Phoenix, and Dileep George. “Teaching Compositionality to CNNs.” *CVPR*, 2017.

[SZ14] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition.” *arXiv preprint arXiv:1409.1556*, 2014.

[SZZ18] Alane Suhr, Stephanie Zhou, Iris Zhang, Huajun Bai, and Yoav Artzi. “A corpus for reasoning about natural language grounded in photographs.” *arXiv preprint arXiv:1811.00491*, 2018.

[TB19] Hao Tan and Mohit Bansal. “Lxmert: Learning cross-modality encoder representations from transformers.” *arXiv preprint arXiv:1908.07490*, 2019.

[THD15] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. “Simultaneous deep transfer across domains and tasks.” In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4068–4076, 2015.

[TML14] Kewei Tu, Meng Meng, Mun Wai Lee, Tae Eun Choe, and Song-Chun Zhu. “Joint video and text parsing for understanding events and answering queries.” *IEEE MultiMedia*, **21**(2):42–70, 2014.

[TS20] Ashish V Thapliyal and Radu Soricut. "Cross-modal Language Generation using Pivot Stabilization for Web-scale Language Coverage." *arXiv preprint arXiv:2005.00246*, 2020.

[TZS16] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. "Movieqa: Understanding stories in movies through question-answering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4631–4640, 2016.

[VCS16] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. "Diverse beam search: Decoding diverse solutions from neural sequence models." *arXiv preprint arXiv:1610.02424*, 2016.

[VK19] Arnaud Van Looveren and Janis Klaise. "Interpretable Counterfactual Explanations Guided by Prototypes." *arXiv preprint arXiv:1907.02584*, 2019.

[VKK19] Ville Vakkuri, Kai-Kristian Kemell, Joni Kultanen, Mikko Siponen, and Pekka Abrahamsson. "Ethically Aligned Design of Autonomous Systems: Industry viewpoint and an empirical study." *arXiv preprint arXiv:1906.07946*, 2019.

[VLP15] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. "Cider: Consensus-based image description evaluation." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

[VSP17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is All you Need." In *NIPS*, 2017.

[WKR16] Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. "Axis: Generating explanations at scale with learnersourcing and machine learning." In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pp. 379–388. ACM, 2016.

[WMR17] Sandra Wachter, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." *Harvard Journal of Law & Technology*, **31**(2):2018, 2017.

[WRV16] Tong Wang, Cynthia Rudin, Finale Velez-Doshi, Yimin Liu, Erica Klampfl, and Perry MacNeille. "Bayesian rule sets for interpretable classification." In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1269–1274. IEEE, 2016.

[WYT17] Tong Wang, Xingdi Yuan, and Adam Trischler. "A joint model for question answering and question generation." *arXiv preprint arXiv:1706.01450*, 2017.

[WZ11] Tianfu Wu and Song-Chun Zhu. "A numerical study of the bottom-up and top-down inference processes in and-or graphs." *International journal of computer vision*, **93**(2):226–252, 2011.

[XWY20] Xing Xu, Tan Wang, Yang Yang, Alan Hanjalic, and Heng Tao Shen. "Radial Graph Convolutional Network for Visual Question Generation." *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[YGL16] Shaohua Yang, Qiaozi Gao, Changsong Liu, Caiming Xiong, Song-Chun Zhu, and Joyce Y. Chai. "Grounded Semantic Role Labeling." In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 149–159, San Diego, California, 2016. Association for Computational Linguistics.

[YGS18] Shaohua Yang, Qiaozi Gao, Sari Saba-Sadiya, and Joyce Chai. "Commonsense Justification for Action Explanation." In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2627–2637, 2018.

[YLH14] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions." *Transactions of the Association for Computational Linguistics*, **2**:67–78, 2014.

[YLL18] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. "Visual curiosity: Learning to ask questions to learn visual recognition." *arXiv preprint arXiv:1810.00912*, 2018.

[YLS18] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. "Mattnet: Modular attention network for referring expression comprehension." In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1307–1315, 2018.

[YWG18] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding." In *Advances in neural information processing systems*, pp. 1031–1042, 2018.

[ZAR14] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. "Capturing long-tail distributions of object subcategories." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 915–922, 2014.

[ZBF19] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. "From recognition to cognition: Visual commonsense reasoning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6720–6731, 2019.

[ZCN17] Quanshi Zhang, Ruiming Cao, Ying Nian Wu, and Song-Chun Zhu. “Mining object parts from cnns via active question-answering.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 346–355, 2017.

[ZF14] Matthew D Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks.” In *European conference on computer vision*, pp. 818–833. Springer, 2014.

[ZGB16] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. “Visual7w: Grounded question answering in images.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4995–5004, 2016.

[ZHB19] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. “HellaSwag: Can a Machine Really Finish Your Sentence?” In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, 2019. Association for Computational Linguistics.

[ZKL16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. “Learning deep features for discriminative localization.” In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pp. 2921–2929. IEEE, 2016.

[ZM07] Song-Chun Zhu, David Mumford, et al. “A stochastic grammar of images.” *Foundations and Trends® in Computer Graphics and Vision*, **2**(4):259–362, 2007.

[ZMJ16] Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. “DeepRED–Rule extraction from deep neural networks.” In *International Conference on Discovery Science*, pp. 457–473. Springer, 2016.

[ZNZ18] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. “Interpretable convolutional neural networks.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8827–8836, 2018.

[ZVF16] Xiangxin Zhu, Carl Vondrick, Charless C Fowlkes, and Deva Ramanan. “Do we need more training data?” *International Journal of Computer Vision*, **119**(1):76–92, 2016.

[ZWZ18] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. “Interpretable Convolutional Neural Networks.” *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8827–8836, 2018.

[ZYM19] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. “Interpreting cnns via decision trees.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6261–6270, 2019.

[ZYY18] Quanshi Zhang, Yu Yang, Qian Yu, and Ying Nian Wu. “Network Transplanting.”, 2018.

[ZZ13] Mingtian Zhao and Song-Chun Zhu. “Abstract painting with interactive control of perceptual entropy.” *ACM Transactions on Applied Perception (TAP)*, **10**(1):1–21, 2013.

[ZZC13] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. “Fast and accurate shift-reduce constituent parsing.” In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 434–443, 2013.